System Identification in the Short-Time Fourier Transform Domain

Yekutiel (Kuti) Avargel

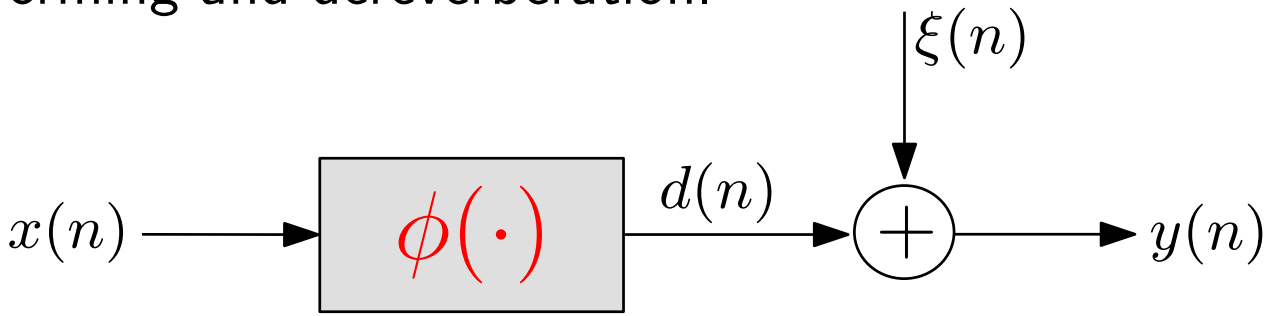
Electrical Engineering Department
Technion - Israel Institute of Technology

Supervised by: Prof. Israel Cohen

Outline

Introduction

- **Identification of systems** is a fundamental problem in many practical applications, including acoustic echo cancellation, beamforming and dereverberation.

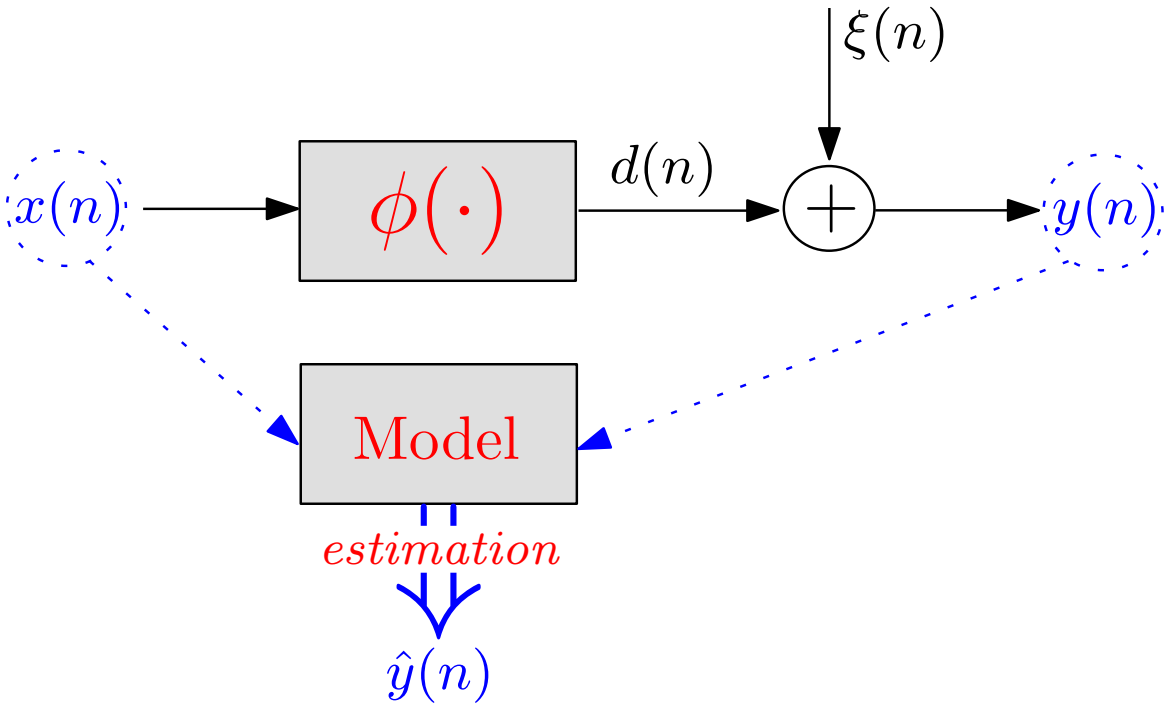


$$y(n) = \{\phi x\}(n) + \xi(n) = d(n) + \xi(n)$$

Introduction (cont.)

System identification problem:

Given $\{x(n), y(n)\}$, **construct a model** and select its parameters so that the model output $\hat{y}(n)$ **best estimates** the signal $y(n)$.



Time-domain identification

- Assume the model output depends **linearly** on its coefficients:

$$\hat{y}(n) = \mathbf{x}^T(n)\boldsymbol{\theta}$$

- Batch- and adaptive-estimation approaches are employed:
  - $\hat{\theta}_{\text{LS}} = \left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H\mathbf{y}$
  - $\hat{\theta}_{\text{MSE}} = \left[E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\}\right]^{-1} E\left\{\mathbf{x}(n)y(n)\right\}$
  - $\hat{\theta}_{\text{LMS}}(n+1) = \hat{\theta}_{\text{LMS}}(n) + \mu e(n)\mathbf{x}(n)$

When $\dim\theta$ is large, time-domain approaches suffer from extremely high computational complexity and slow convergence.

Time-domain identification

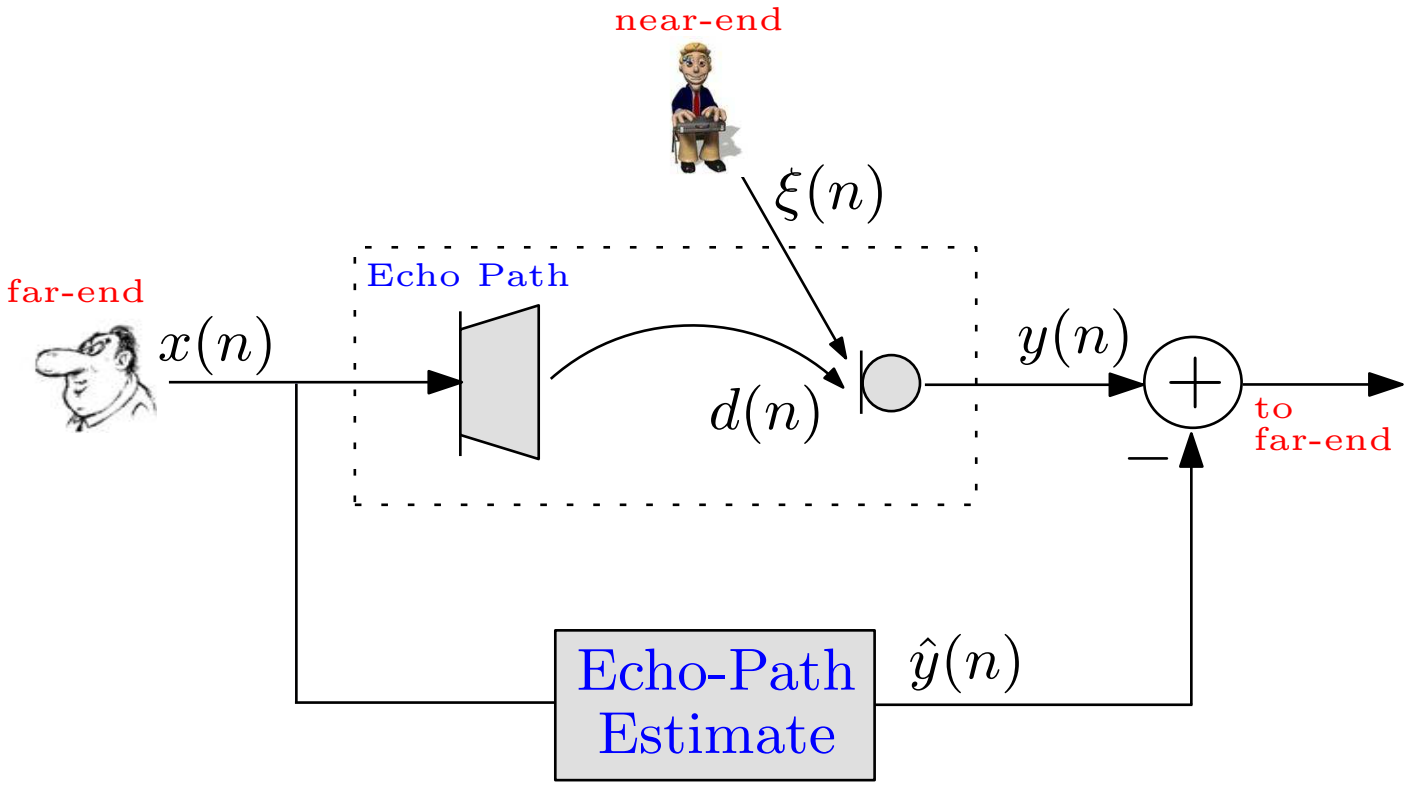
- Assume the model output depends **linearly** on its coefficients:

$$\hat{y}(n) = \mathbf{x}^T(n)\boldsymbol{\theta}$$

- Batch- and adaptive-estimation approaches are employed:
  - $\hat{\theta}_{\text{LS}} = \left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H\mathbf{y}$
  - $\hat{\theta}_{\text{MSE}} = \left[E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\}\right]^{-1} E\left\{\mathbf{x}(n)y(n)\right\}$
  - $\hat{\theta}_{\text{LMS}}(n+1) = \hat{\theta}_{\text{LMS}}(n) + \mu e(n)\mathbf{x}(n)$

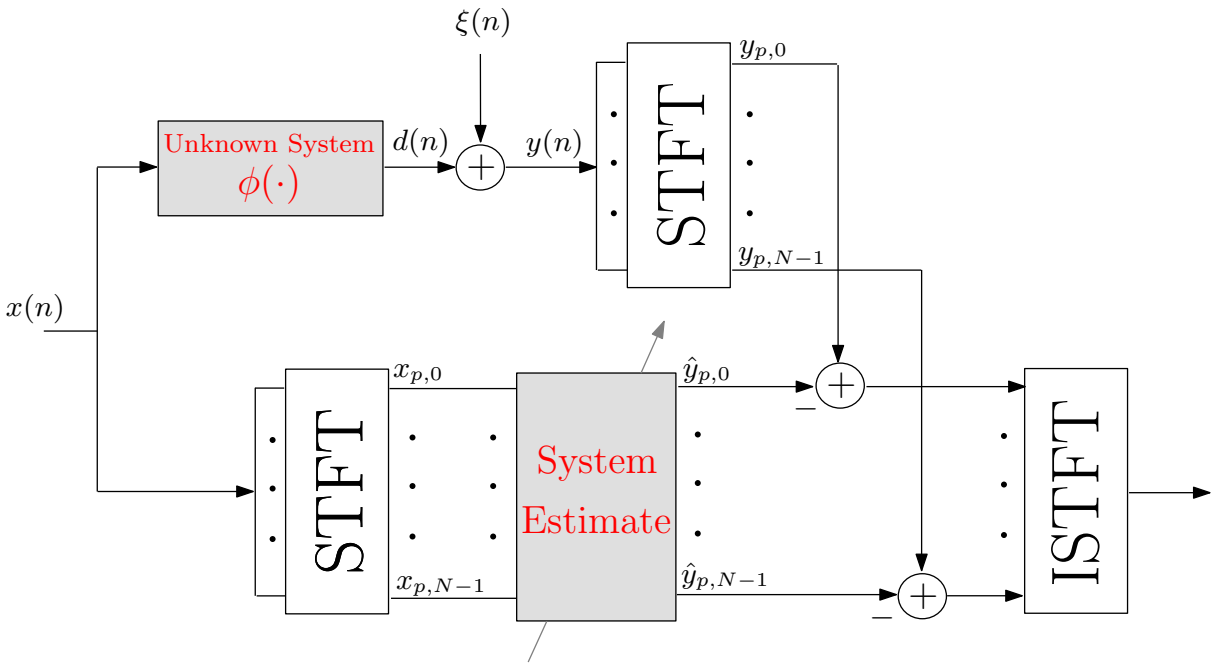
When $\dim\boldsymbol{\theta}$ is large, time-domain approaches suffer from extremely high computational complexity and slow convergence.

Acoustic echo cancellation



Subband identification

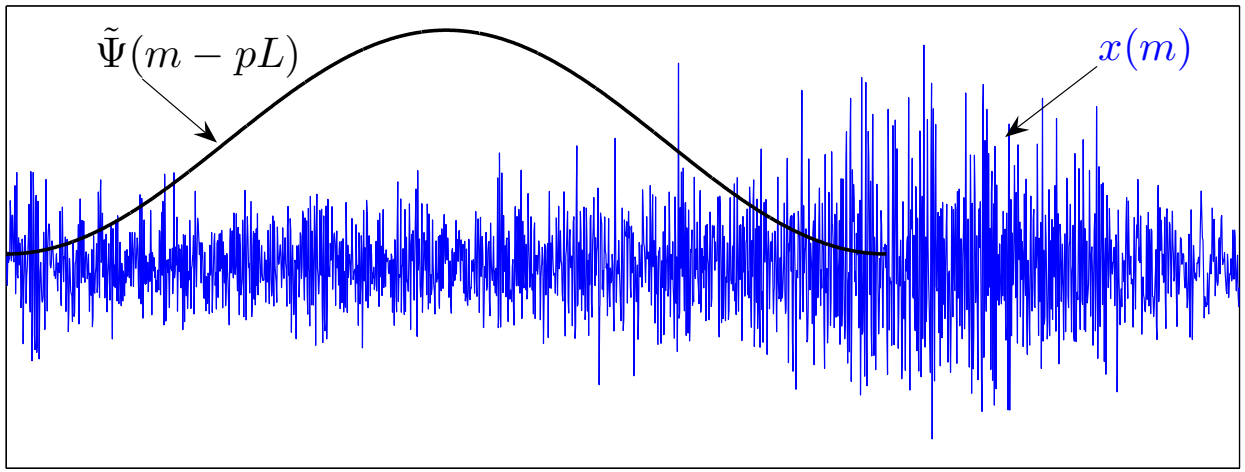
- Alternatively, **subband (multirate) techniques** are used for improved system identification.
- Computational efficiency and improved convergence rate is achieved due to processing in distinct subbands.



Short-Time Fourier Transform (STFT)

- The STFT representation of a signal $x(n)$ is given by

$$x_{p,k} = \sum_m x(m)\tilde{\psi}^*(m - pL)e^{-j\frac{2\pi}{N}k(m-pL)}$$



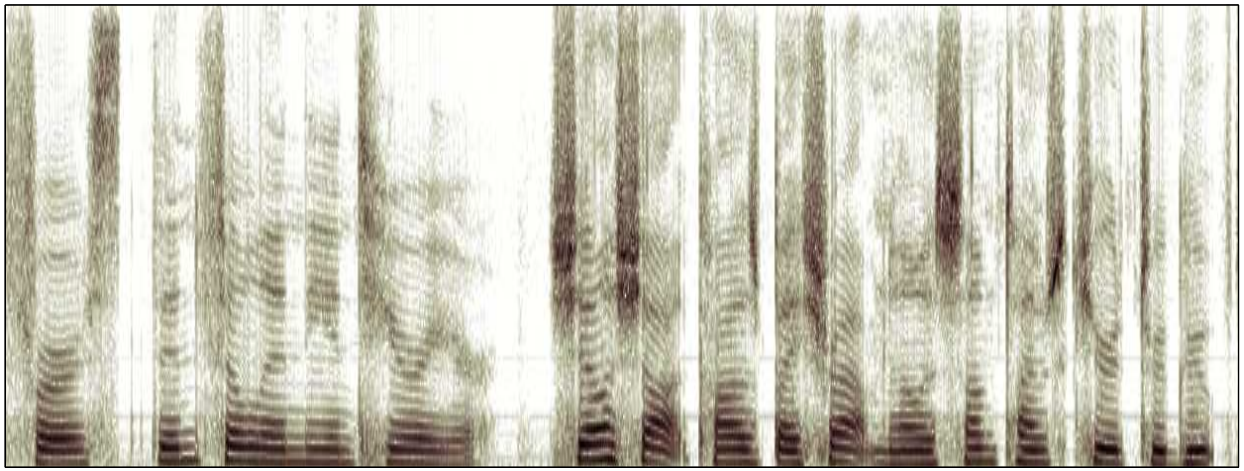
- The inverse STFT (ISTFT) is given by

$$x(n) = \sum_p \sum_{k=0}^{N-1} x_{p,k}\psi(n - pL)e^{j\frac{2\pi}{N}k(n-pL)}$$

Short-Time Fourier Transform (STFT)

- The STFT representation of a signal $x(n)$ is given by

$$x_{p,k} = \sum_m x(m)\tilde{\psi}^*(m - pL)e^{-j\frac{2\pi}{N}k(m-pL)}$$



- The inverse STFT (ISTFT) is given by

$$x(n) = \sum_p \sum_{k=0}^{N-1} x_{p,k}\psi(n - pL)e^{j\frac{2\pi}{N}k(n-pL)}$$

Research objectives

How can the system $\phi(\cdot)$ be represented and estimated in the STFT domain?

The following two cases will be considered:

- $\phi(\cdot)$ is a linear system.
- $\phi(\cdot)$ is a nonlinear system.

Linear Systems in the STFT Domain

Linear system identification

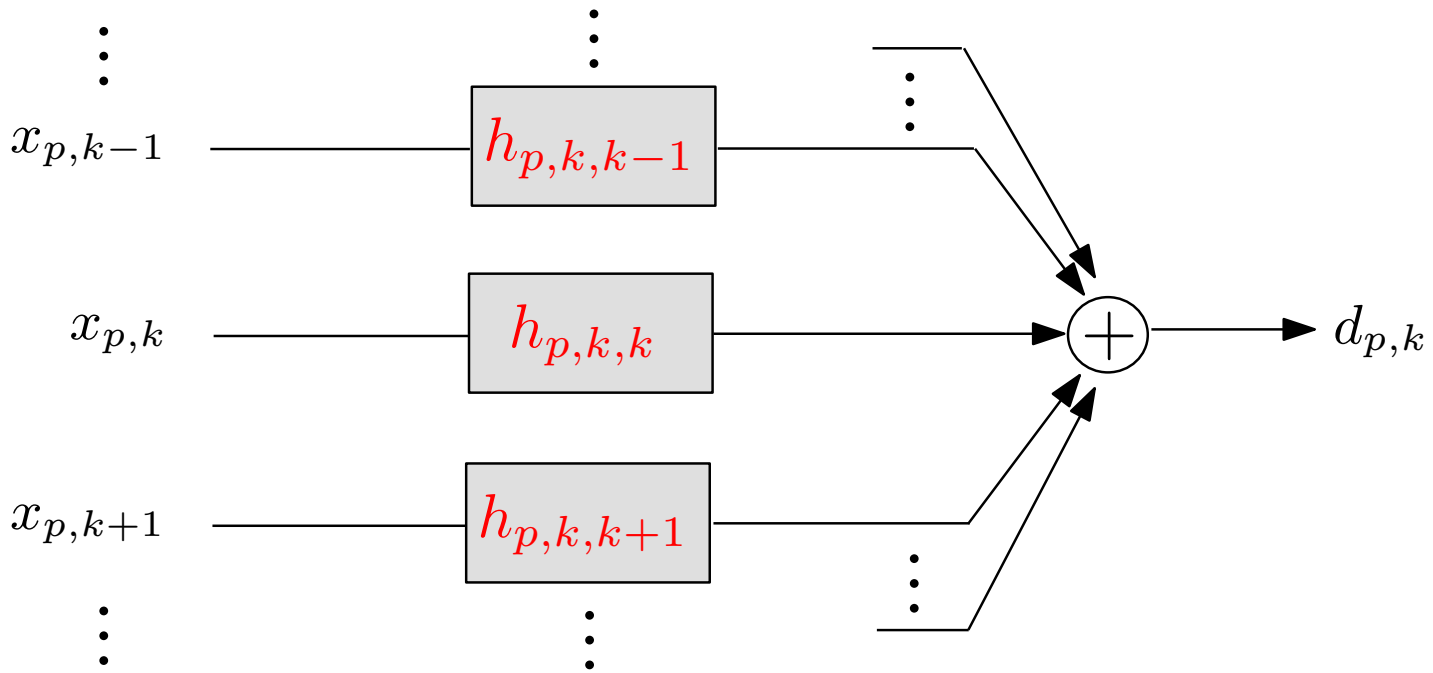
- In the linear (time-invariant) case $d(n) = h(n) * x(n)$, where $h(n)$ is the system impulse response.
- To perfectly represent $h(n)$ in the STFT domain crossband filters between subbands are generally required:

$$d_{p,k} = \sum_{k'=0}^{N-1} \sum_{p'=0}^{M-1} x_{p-p',k'} \, h_{p',k,k'}$$

where $h_{p',k,k'}$ is the crossband filter from frequency-bin k' to frequency-bin k.

Crossband filters

- **Crossband filters illustration:**



Crossband filters (cont.)

- The crossband filter $h_{p,k,k'}$ depends on both $h(n)$ and the STFT analysis/ synthesis parameters:

$$h_{p,k,k'} = \left\{ h(n) * \phi_{k,k'}(n) \right\} \Big|_{n=pL}$$

where

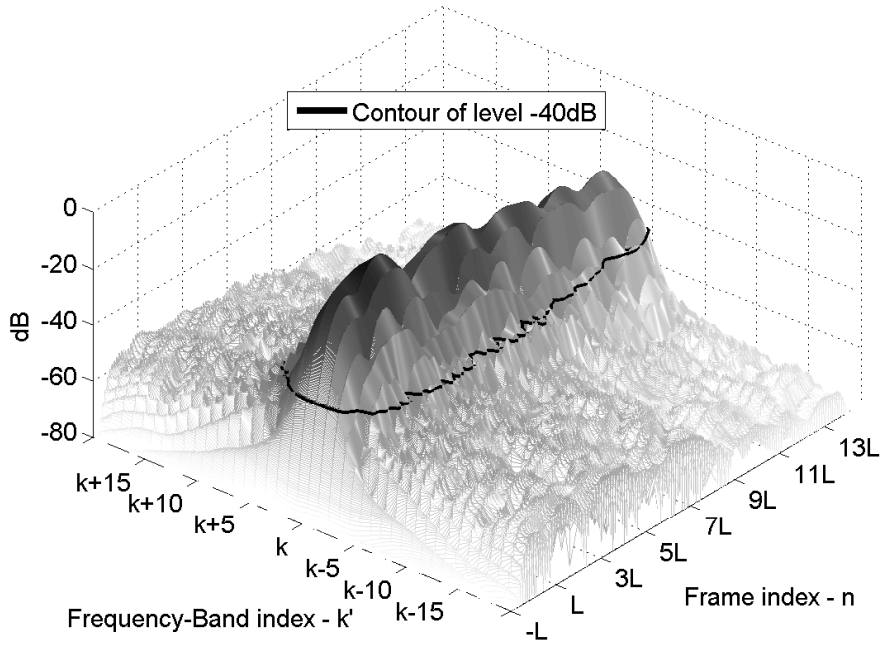
$$\Phi_{k,k'}(\theta) = \tilde{\Psi}\left(\theta - \frac{2\pi}{N}k\right) \Psi\left(\theta - \frac{2\pi}{N}k'\right)$$

and $\tilde{\Psi}(\theta)$ and $\Psi(\theta)$ are the DTFT of the analysis and synthesis windows, respectively [Avargel & Cohen, 07'].

- For fixed k and k', the filter $h_{p,k,k'}$ is **noncausal** in general, with $\left\lceil \frac{N}{L} \right\rceil - 1$ noncausal coefficients.

Crossband filters (cont.)

Practically, relatively few crossband filters need to be considered.



A mesh plot of the crossband filters $|h_{p,1,k'}|$ for a synthetic impulse response.
L denotes the decimation factor.

For system identification in the STFT domain:

An estimator for the crossband filters is required.

Motivation

- Utilizing crossband filters between the subbands is inferior to either fullband adaptive algorithms or subband approaches that does not include crossband filters [Gilloire et al. 92'].
- Most applications **disregard** the crossband filters in the subband identification process.

An open question still remains:

Why does the inclusion of crossband filters worsen the performance of subband system identification algorithms?

Motivation

- Utilizing crossband filters between the subbands is inferior to either fullband adaptive algorithms or subband approaches that does not include crossband filters [Gilloire et al. 92'].
- Most applications **disregard** the crossband filters in the subband identification process.

An open question still remains:

Why does the inclusion of crossband filters worsen the performance of subband system identification algorithms?

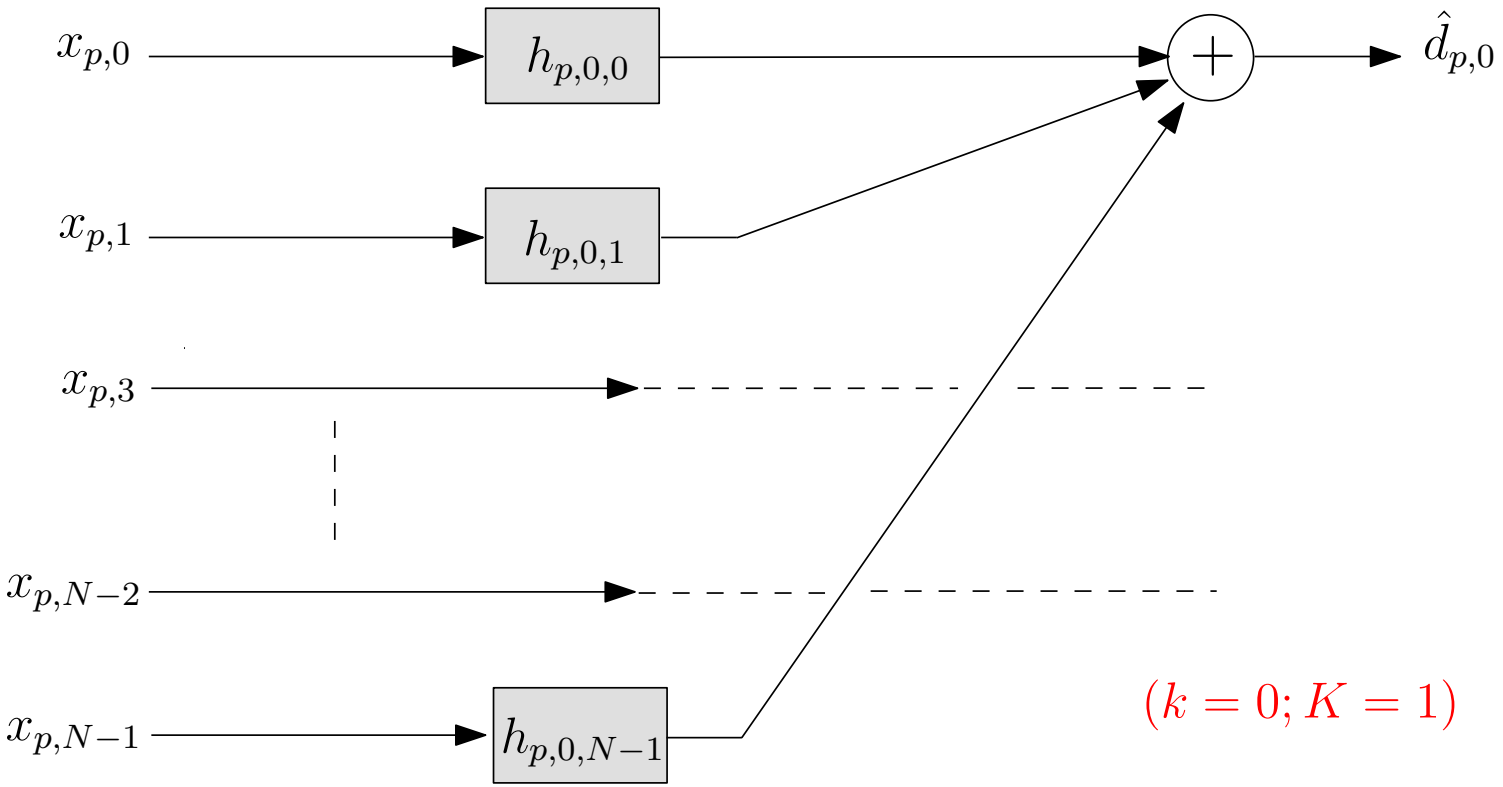
System identification using crossband filters

- Let $\hat{y}_{p,k}$ be the resulting estimate of $y_{p,k}$ using only $2K+1$ crossband filters around the frequency-band k:

System $$y_{p,k} = \sum_{k'=0}^{N-1} \sum_{p'=0}^{M-1} x_{p-p',k'} \bar{h}_{p',k,k'} + \xi_{p,k}$$

Model $$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} \sum_{p'=0}^{M-1} x_{p-p',k'} h_{p',k,k'}$$

Crossband filters periodicity



Batch estimation of crossband filters

- Let $\mathbf{y}_k = \left[\begin{array}{cccc} y_{0,k} & y_{1,k} & \cdots & y_{P-1,k} \end{array} \right]^T$ denote a time-trajectory of y_{pk} at frequency-bin k.
- Let $\boldsymbol{\theta}_k$ be the model parameter vector at frequency-bin k, consisting of $2K+1$ crossband filters.
- Let $\boldsymbol{\Delta}_k$ be a concatenation of the input Toeplitz matrices:

$$\boldsymbol{\Delta}_k = \left[\begin{array}{cccccc} \mathbf{X}_{(k-K) \bmod N} & \mathbf{X}_{(k-K+1) \bmod N} & \cdots & \cdots & \mathbf{X}_{(k+K) \bmod N} \end{array} \right]$$

The output signal estimate in a vector form:

$$\hat{\mathbf{y}}_k \left(\boldsymbol{\theta}_k \right) = \boldsymbol{\Delta}_k \boldsymbol{\theta}_k$$

Batch estimation of crossband filters

- Let $\mathbf{y}_k = \left[\begin{array}{cccc} y_{0,k} & y_{1,k} & \cdots & y_{P-1,k} \end{array} \right]^T$ denote a time-trajectory of y_{pk} at frequency-bin k.
- Let $\boldsymbol{\theta}_k$ be the model parameter vector at frequency-bin k, consisting of $2K+1$ crossband filters.
- Let $\boldsymbol{\Delta}_k$ be a concatenation of the input Toeplitz matrices:

$$\boldsymbol{\Delta}_k = \left[\begin{array}{cccccc} \mathbf{X}_{(k-K) \bmod N} & \mathbf{X}_{(k-K+1) \bmod N} & \cdots & \cdots & \mathbf{X}_{(k+K) \bmod N} \end{array} \right]$$

The output signal estimate in a vector form:

$$\hat{\mathbf{y}}_k \left(\boldsymbol{\theta}_k \right) = \boldsymbol{\Delta}_k \boldsymbol{\theta}_k$$

Least-squares (LS) estimate

- LS optimization problem:

$$\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}_k} \|\mathbf{y}_k - \boldsymbol{\Delta}_k \boldsymbol{\theta}_k\|^2$$

LS estimate:

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{\Delta}_k^H \boldsymbol{\Delta}_k\right)^{-1} \boldsymbol{\Delta}_k^H \mathbf{y}_k$$

MSE analysis

- The (normalized) mse is defined by

$$\epsilon_k(K) = \frac{E\left\{\left\|\mathbf{d}_k - \hat{\mathbf{y}}_k\left(\hat{\boldsymbol{\theta}}_k\right)\right\|^2\right\}}{E\left\{\left\|\mathbf{d}_k\right\|^2\right\}}$$

- **Assumption 1:** $x_{p,k}$ and $\xi_{p,k}$ are zero-mean **white** Gaussian complex-valued signals with variance σ_x^2 and σ_ξ^2.
- **Assumption 2:** $x_{p,k}$ and $\xi_{p,k}$ are **statistically independent**.

MSE analysis (cont.)

- The mse can be rewritten as:

$$\epsilon_k(K) = 1 + \epsilon_1 - \epsilon_2$$

$$\epsilon_1 = \frac{1}{E\left\{\|\mathbf{d}_k\|^2\right\}} E\left\{\boldsymbol{\xi}_k^H \boldsymbol{\Delta}_k \left(\boldsymbol{\Delta}_k^H \boldsymbol{\Delta}_k\right)^{-1} \boldsymbol{\Delta}_k^H \boldsymbol{\xi}_k\right\}$$

$$\epsilon_2 = \frac{1}{E\left\{\|\mathbf{d}_k\|^2\right\}} E\left\{\mathbf{d}_k^H \boldsymbol{\Delta}_k \left(\boldsymbol{\Delta}_k^H \boldsymbol{\Delta}_k\right)^{-1} \boldsymbol{\Delta}_k^H \mathbf{d}_k\right\}$$

- Under the whiteness and independence assumptions, we get

$$\epsilon_1 = \frac{\sigma_\xi^2 M(2K+1)}{\sigma_x^2 P \|\mathbf{h}_k\|^2}.$$

MSE analysis (cont.)

- The mse can be rewritten as:

$$\epsilon_k(K) = 1 + \epsilon_1 - \epsilon_2$$

$$\epsilon_1 = \frac{1}{E\left\{\|\mathbf{d}_k\|^2\right\}} E\left\{\boldsymbol{\xi}_k^H \boldsymbol{\Delta}_k \left(\boldsymbol{\Delta}_k^H \boldsymbol{\Delta}_k\right)^{-1} \boldsymbol{\Delta}_k^H \boldsymbol{\xi}_k\right\}$$

$$\epsilon_2 = \frac{1}{E\left\{\|\mathbf{d}_k\|^2\right\}} E\left\{\mathbf{d}_k^H \boldsymbol{\Delta}_k \left(\boldsymbol{\Delta}_k^H \boldsymbol{\Delta}_k\right)^{-1} \boldsymbol{\Delta}_k^H \mathbf{d}_k\right\}$$

- Under the whiteness and independence assumptions, we get

$$\epsilon_1 = \frac{\sigma_\xi^2 M\left(2K+1\right)}{\sigma_x^2 P \left\|\mathbf{h}_k\right\|^2}.$$

MSE analysis (cont.)

Assumption 3: $x_{p,k}$ is **variance-ergodic**:

$$\frac{1}{P}\sum_{p=0}^{P-1} x_{p,k}x_{p+s,k'}^{*} \approx E\left\{x_{p,k}x_{p+s,k'}^{*}\right\}.$$

- Consequently,

$$\left(\mathbf{\Delta}_k^H \mathbf{\Delta}_k\right)_{m,\ell} \approx P\sigma_x^2\delta(\ell-m).$$

and ϵ_2 reduces to

$$\epsilon_2 = \frac{1}{\sigma_x^4 P^2 \left\|\mathbf{h}_k\right\|^2}\mathbf{h}_k^H E\left\{\mathbf{\Delta}_k^H\tilde{\mathbf{\Delta}}_k\tilde{\mathbf{\Delta}}_k^H\mathbf{\Delta}_k\right\}\mathbf{h}_k$$

MSE analysis (cont.)

- Let $\eta = \sigma_x^2/\sigma_\xi^2$ denote the SNR. Then, we obtain

MMSE in the k-th frequency bin:

$$\epsilon_k(K) = \frac{\alpha_k(K)}{\eta} + \beta_k(K)$$

[Avargel & Cohen, IEEE Trans. Audio, Speech, Language Process., 07']

$$\alpha_k(K) \triangleq \frac{M}{P\left\|\bar{\mathbf{h}}_k\right\|^2}(2K+1)$$

$$\beta_k(K) \triangleq 1 - \frac{M(2K+1)}{P} - \frac{1}{\left\|\bar{\mathbf{h}}_k\right\|^2}\sum_{m=0}^{2K}\left\|\bar{\mathbf{h}}_{k,(k-K+m) \bmod N}\right\|^2$$

MSE analysis (cont.)

- Let $\eta = \sigma_x^2/\sigma_\xi^2$ denote the SNR. Then, we obtain

MMSE in the k-th frequency bin:

$$\epsilon_k(K) = \frac{\alpha_k(K)}{\eta} + \beta_k(K)$$

[Avargel & Cohen, IEEE Trans. Audio, Speech, Language Process., 07']

$$\alpha_k(K) \triangleq \frac{M}{P\left\|\bar{\mathbf{h}}_k\right\|^2}(2K+1)$$

$$\beta_k(K) \triangleq 1 - \frac{M(2K+1)}{P} - \frac{1}{\left\|\bar{\mathbf{h}}_k\right\|^2}\sum_{m=0}^{2K}\left\|\bar{\mathbf{h}}_{k,(k-K+m) \bmod N}\right\|^2$$

MSE analysis (cont.)

- The resulting mmse satisfies

$$\epsilon_k(K+1) > \epsilon_k(K) \quad \text{for} \quad \eta \to 0 \ \text{(low SNR)}$$

$$\epsilon_k(K+1) \leq \epsilon_k(K) \quad \text{for} \quad \eta \to \infty \ \text{(high SNR)}$$

- Let $\eta_k\left(K+1 \to K\right)$ denote the SNR-intersection point of the curves $\epsilon_k(K)$ and $\epsilon_k(K+1)$.

$$\eta_k\left(K \to K-1\right) \leq \eta_k\left(K+1 \to K\right)$$

$$\eta_k\left(K+1 \to K\right) \propto \frac{1}{P}$$

(P is the length of $x_{p,k}$ in frequency-bin k)

MSE analysis (cont.)

- The resulting mmse satisfies

$$\epsilon_k(K+1) > \epsilon_k(K) \quad \text{for} \quad \eta \to 0 \ \text{(low SNR)}$$

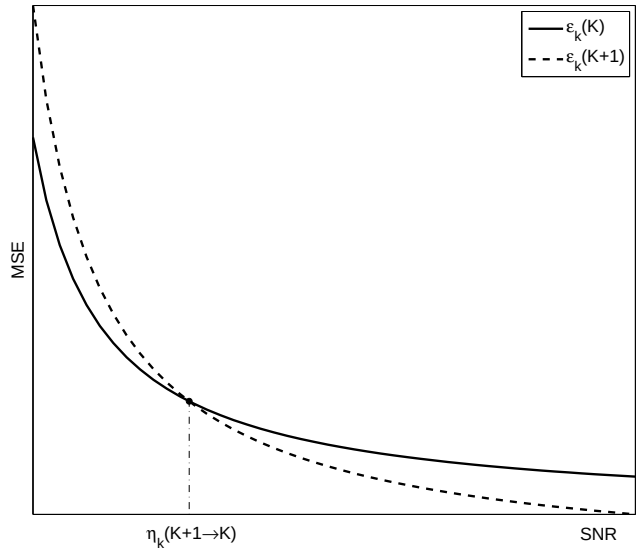
$$\epsilon_k(K+1) \leq \epsilon_k(K) \quad \text{for} \quad \eta \to \infty \ \text{(high SNR)}$$

- Let $\eta_k\left(K+1 \to K\right)$ denote the SNR-intersection point of the curves $\epsilon_k(K)$ and $\epsilon_k(K+1)$.

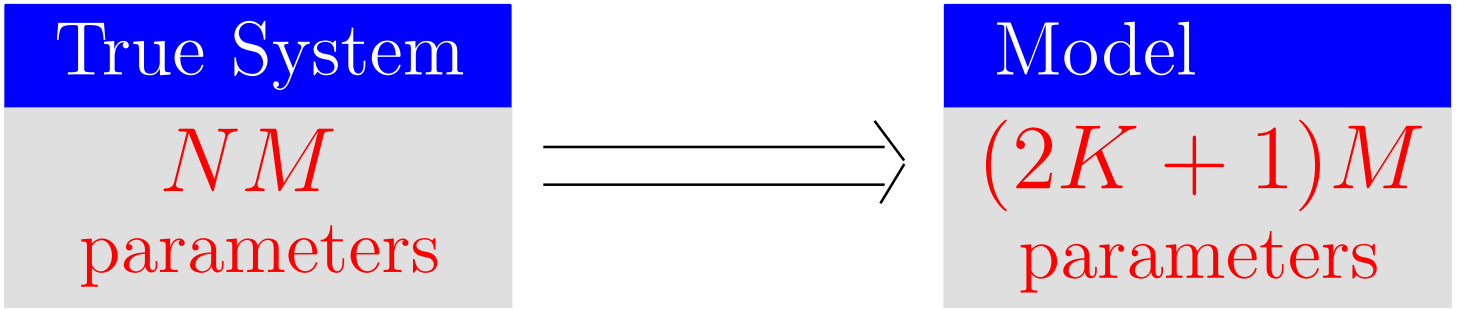
$$\eta_k\left(K \to K-1\right) \leq \eta_k\left(K+1 \to K\right)$$

$$\eta_k\left(K+1 \to K\right) \propto \frac{1}{P}$$

(P is the length of $x_{p,k}$ in frequency-bin k)



Discussion



- Increasing the number of crossband filters not necessarily implies a lower steady-state mse in subbands.

As the SNR increases or as more data becomes available, additional crossband filters can be estimated and a lower MMSE can be achieved.

Discussion (cont.)

- The expressions derived here are related to the problem of model order selection (Bias/Variance tradeoff) [Akaike, 74'], [Rissanen, 78'].
- In this case, the model order is determined by the number of estimated crossband filters.
- Selecting the optimal model complexity for a given data set is a fundamental problem in many system identification applications.
- As the SNR increases or as more data is employable, the optimal model complexity increases, and correspondingly additional cross-terms can be estimated to achieve lower mse.

Discussion (cont.)

- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

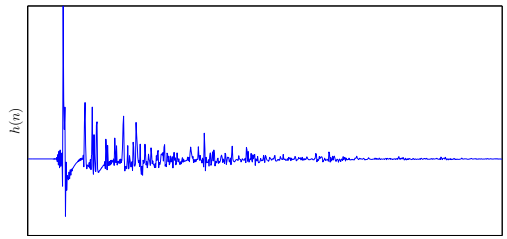
During fast variations - less crossband filters are useful.

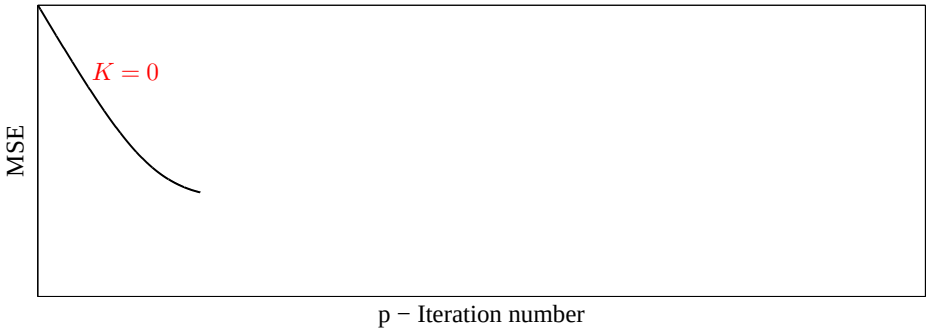
Discussion (cont.)

- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Slow variations:**



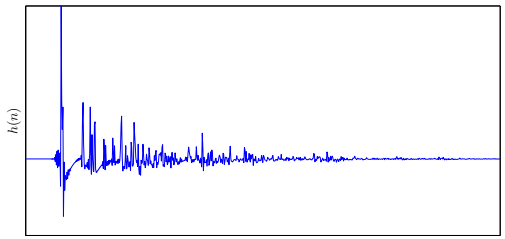


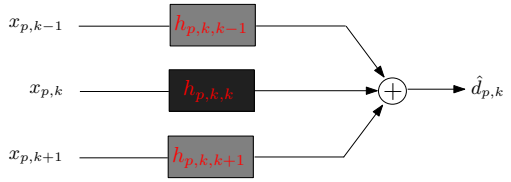
Discussion (cont.)

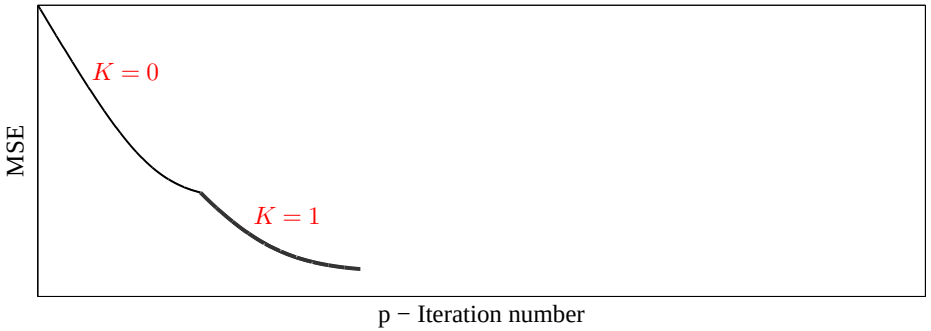
- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Slow variations:**





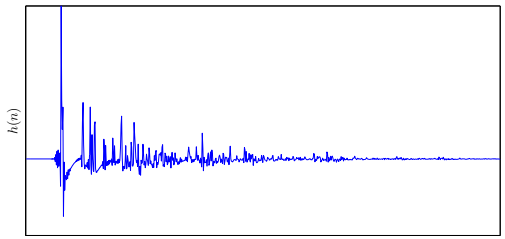


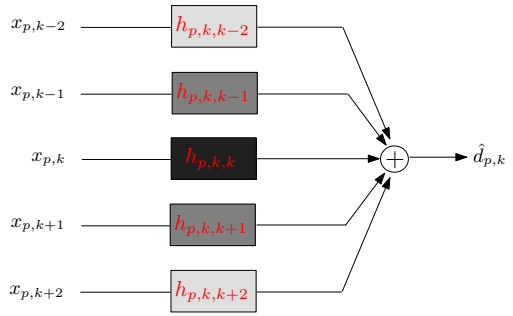
Discussion (cont.)

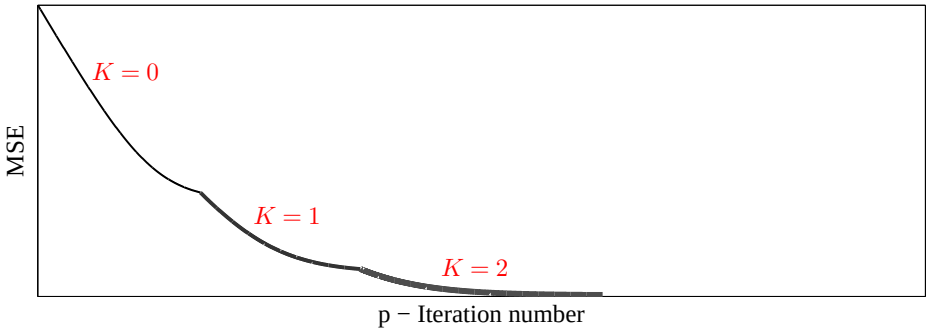
- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Slow variations:**





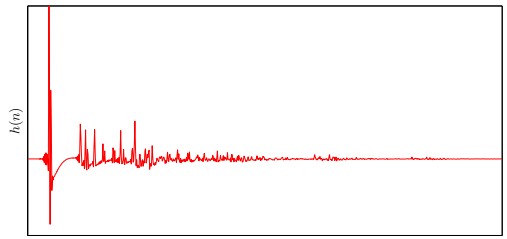


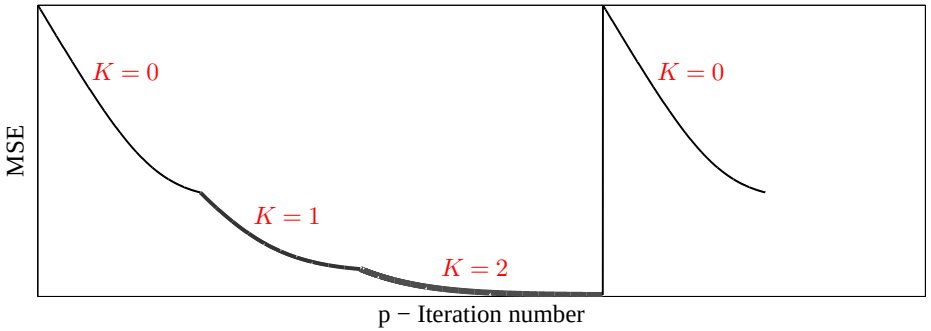
Discussion (cont.)

- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Slow variations:**



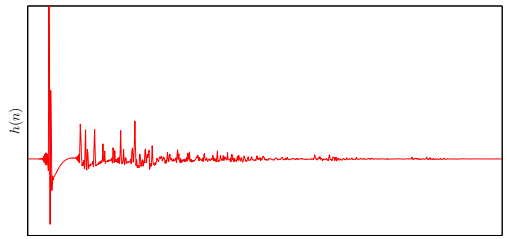


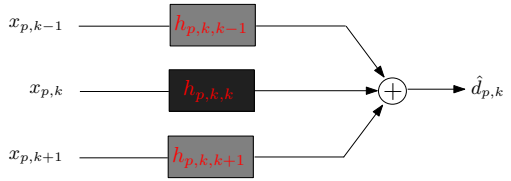
Discussion (cont.)

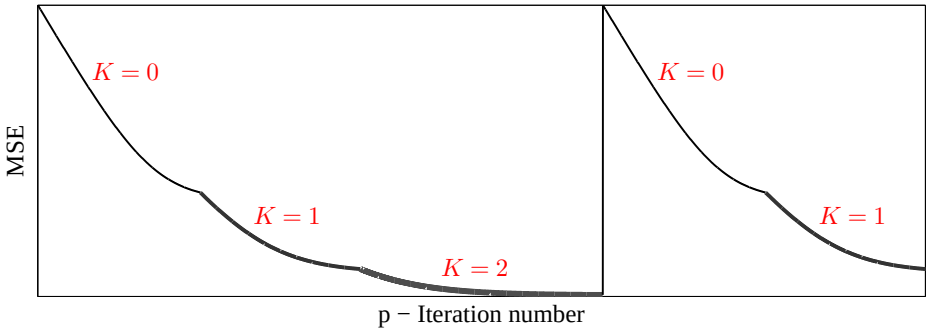
- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Slow variations:**





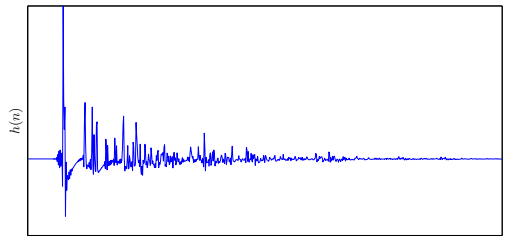


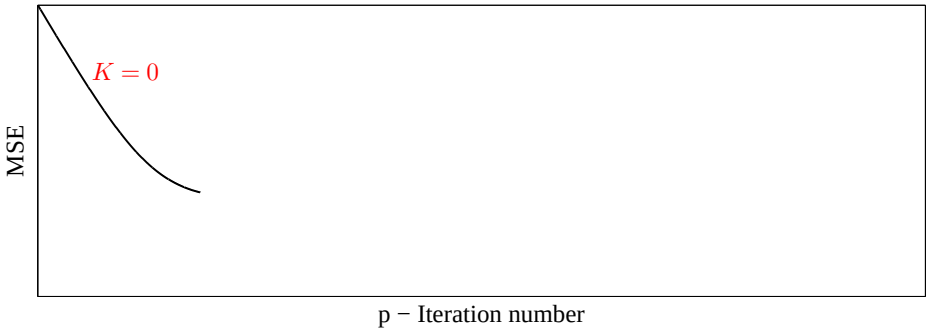
Discussion (cont.)

- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Fast variations:**



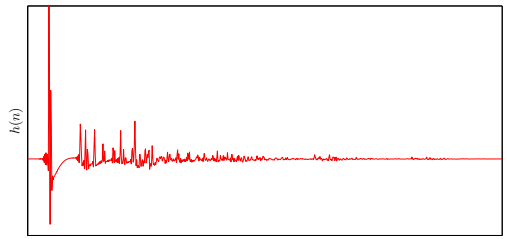


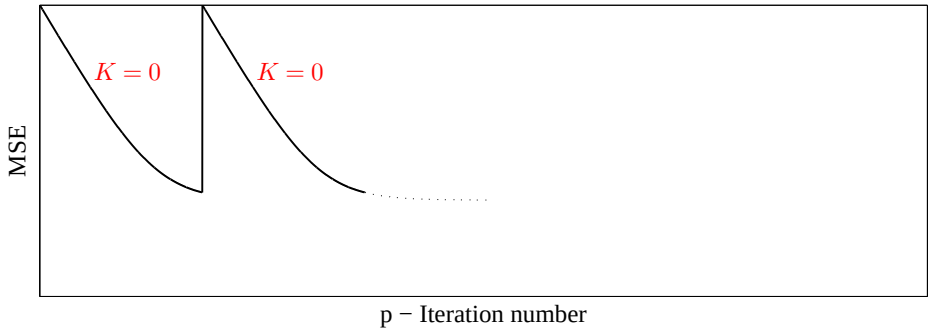
Discussion (cont.)

- The input data length is restricted to enable **tracking capability** during time variations in the impulse response.

During fast variations - less crossband filters are useful.

- **Fast variations:**





Computational complexity

- Let N_h and N_x be the lengths of the system impulse response $h(n)$ and the input signal $x(n)$, respectively.

Complexity of proposed subband approach

$$O_{SB}^{K} = O\left(N_x N_h^2 \frac{N\left(2K+1\right)^2}{L^3}\right)$$

Complexity of fullband approach

$$O_{\mathrm{FB}} = O\left(N_x N_h^2\right)$$

- For $N = 256$, $L = 0.5N$, $N_h = 1500$ and $K = 4$ computational cost is reduced by a factor of 100.

Computational complexity

- Let N_h and N_x be the lengths of the system impulse response $h(n)$ and the input signal $x(n)$, respectively.

Complexity of proposed subband approach

$$O_{SB}^{K} = O\left(N_x N_h^2 \frac{N\left(2K+1\right)^2}{L^3}\right)$$

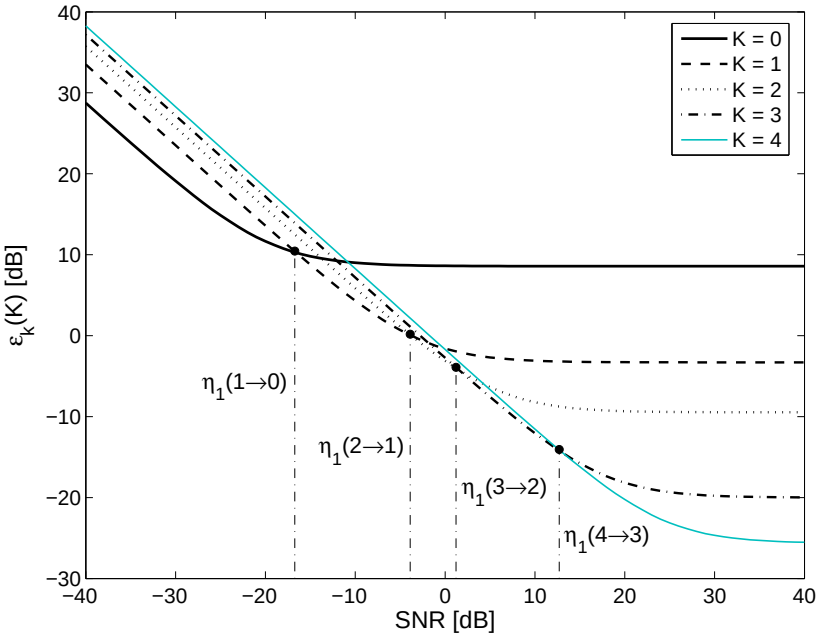
Complexity of fullband approach

$$O_{\mathrm{FB}} = O\left(N_x N_h^2\right)$$

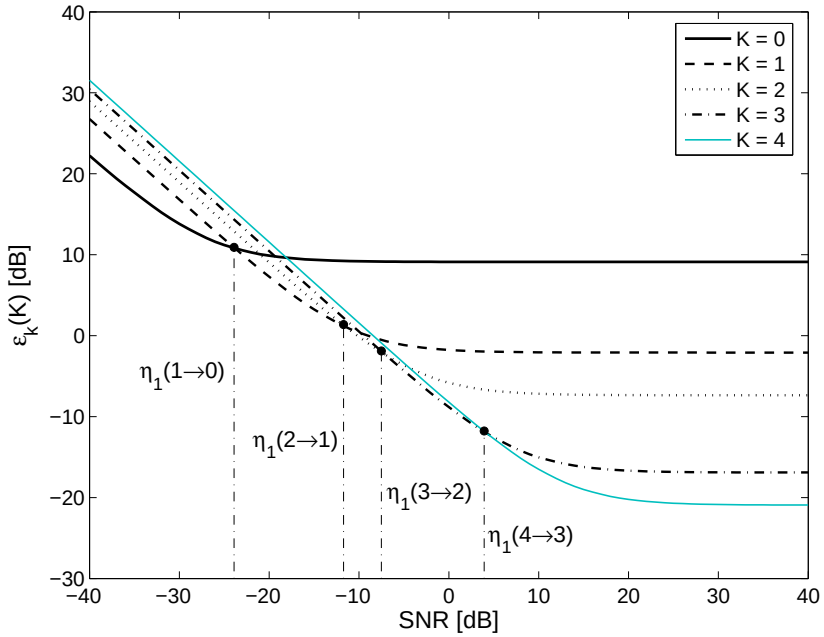
- For $N = 256$, $L = 0.5N$, $N_h = 1500$ and $K = 4$ computational cost is reduced by a factor of 100.

Experimental results

White Gaussian signals ($k = 1$):



(a) $P = 200$



(b) $P = 1000$

Experimental results (cont.)

Acoustic echo cancellation application:

- $x(n)$ is a **speech signal** and the local disturbance $\xi(n)$ consists of a zero-mean white Gaussian local noise.

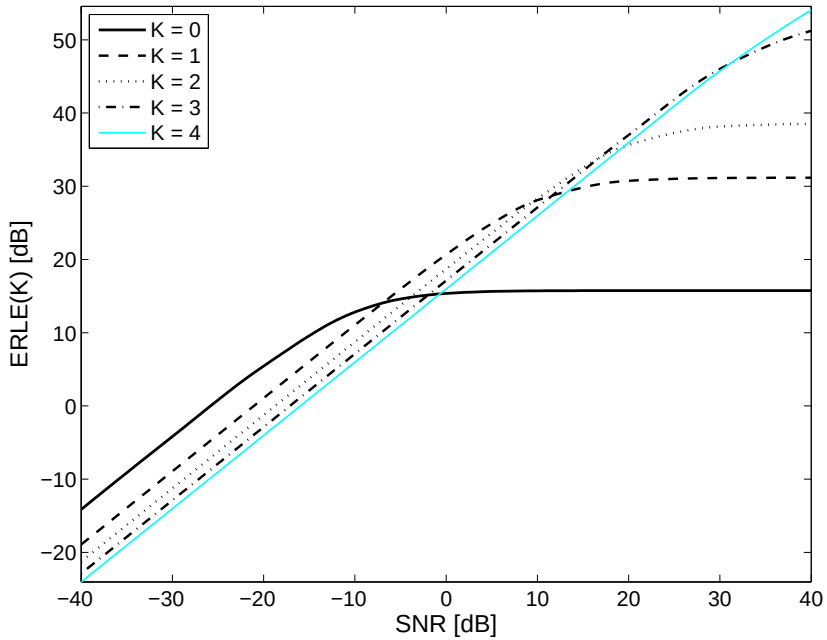
- Performances are evaluated using the echo-return loss enhancement (ERLE):

$$\mathrm{ERLE}(K) = 10 \log \frac{E\left\{d^2(n)\right\}}{E\left\{\left(d(n) - \hat{y}_K(n)\right)^2\right\}}$$

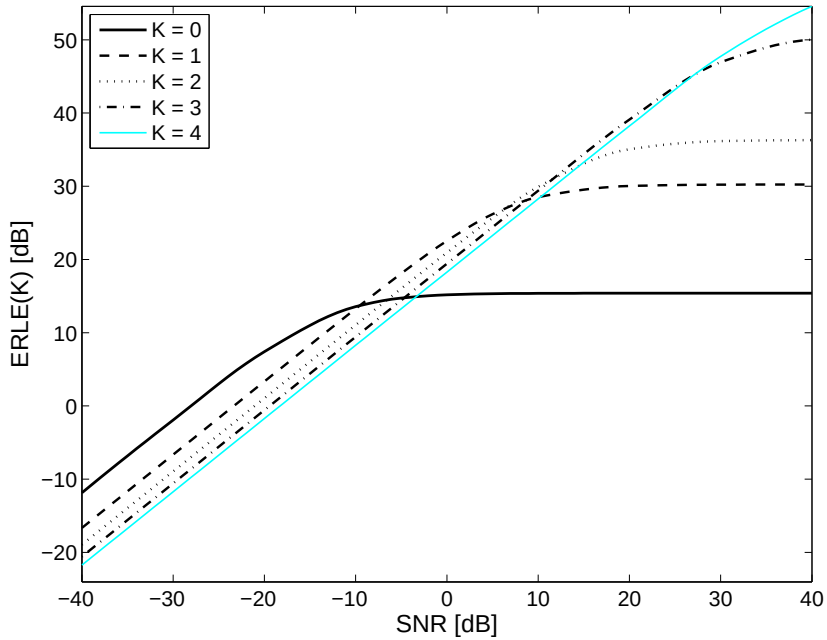
where $\hat{y}_K(n)$ is the inverse STFT of the estimated echo signal using $2K+1$ crossband filters.

Experimental results (cont.)

Acoustic echo cancellation application:



(a) $P = 190$



(b) $P = 322$

The multiplicative transfer function (MTF) approximation

- A widely-used approach to avoid the crossband filters is to approximate the transfer function as **multiplicative** in the STFT domain.
- A relatively l**arge analysis-window length** (N) is assumed.

- **Assumption:** $\tilde{\psi}(n-m)\,h(m) \approx \tilde{\psi}(n)\,h(m)$
- **Approximation:** $d_{p,k} \approx h_k\, x_{p,k}$ $\left(h_k \triangleq \sum_m h(m) e^{-j\frac{2\pi}{N}mk}\right)$

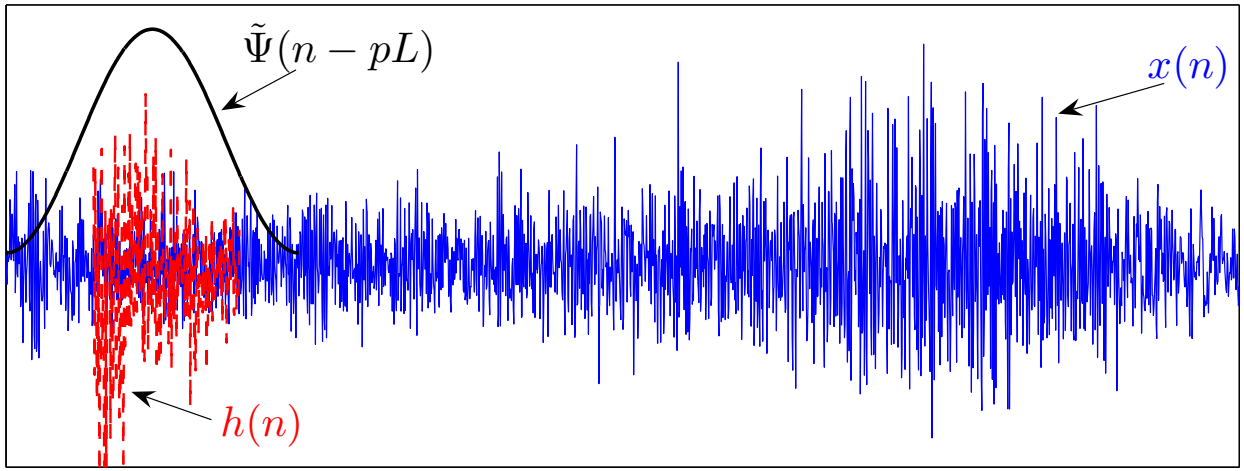
The multiplicative transfer function (MTF) approximation

- A widely-used approach to avoid the crossband filters is to approximate the transfer function as **multiplicative** in the STFT domain.
- A relatively l**arge analysis-window length** (N) is assumed.

- **Assumption:** $\tilde{\psi}(n-m)\,h(m) \approx \tilde{\psi}(n)\,h(m)$
- **Approximation:** $d_{p,k} \approx h_k\, x_{p,k}$ $\qquad \left(h_k \triangleq \sum_m h(m) e^{-j\frac{2\pi}{N}mk}\right)$

The MTF approximation (cont.)

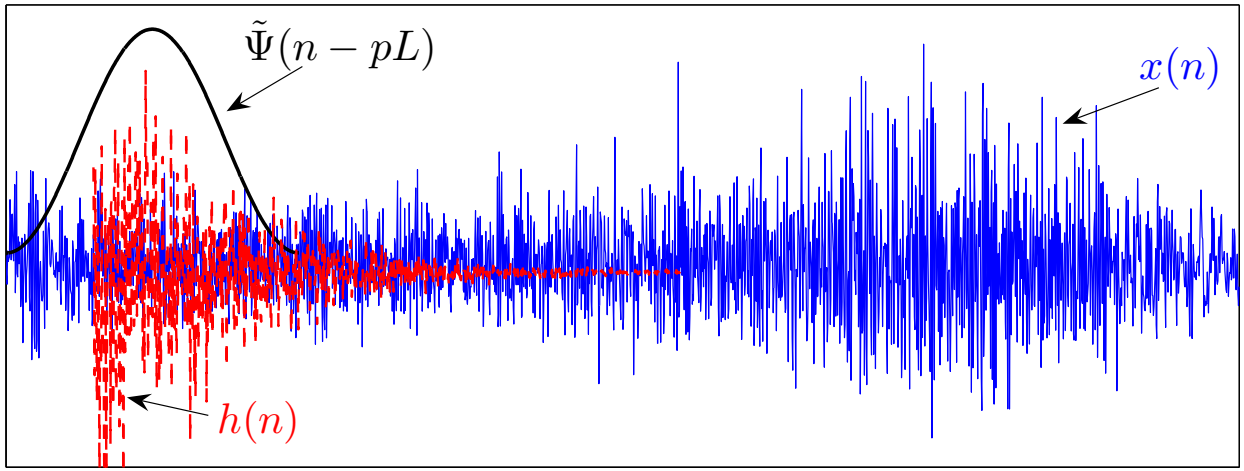
- The MTF approximation becomes more accurate as the analysis window length (N) increases.



The MTF approximation (cont.)

- However, in many applications, $h(n)$ is relatively long.

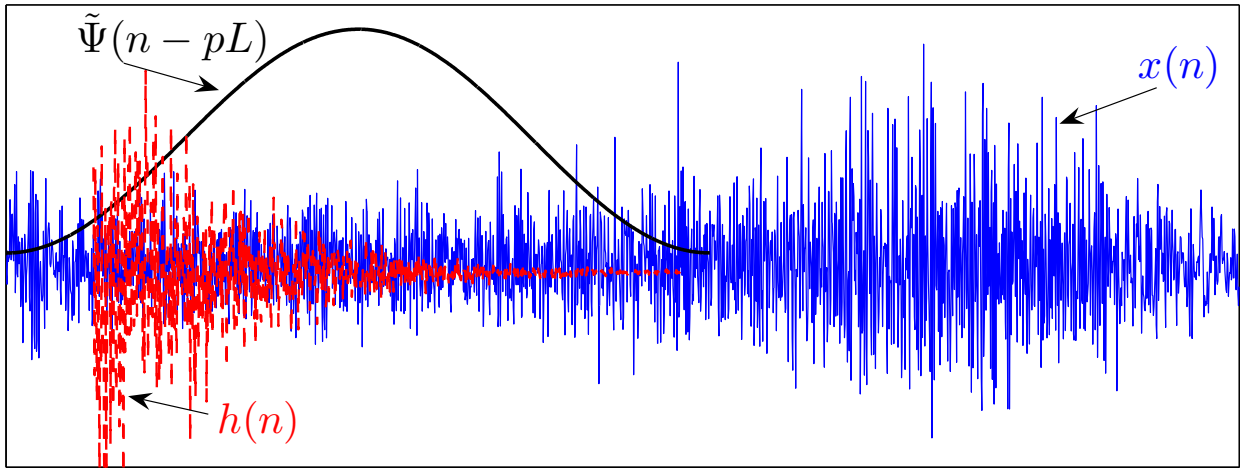
Can we correspondingly **increase** the analysis window length?



The MTF approximation (cont.)

- However, in many applications, $h(n)$ is relatively long.

Can we correspondingly **increase** the analysis window length?



Motivation

- $x(n)$ has finite length $\Longrightarrow$ a fewer number of observations in each frequency bin become available with increasing N.

Trade-off:

$\uparrow N \Longrightarrow$ **More accurate approximation**
$\downarrow N \Longrightarrow$ **Smaller variance of the system estimate**

[Avargel & Cohen, IEEE Signal Process. Lett., 07']

- The mse does not necessarily improve by increasing the length of the analysis window.

There may exist an **optimal window length** that achieves the mmse.

Motivation

- $x(n)$ has finite length $\Longrightarrow$ a fewer number of observations in each frequency bin become available with increasing N.

Trade-off:

$\uparrow N \Longrightarrow$ **More accurate approximation**
$\downarrow N \Longrightarrow$ **Smaller variance of the system estimate**

[Avargel & Cohen, IEEE Signal Process. Lett., 07']

- The mse does not necessarily improve by increasing the length of the analysis window.

There may exist an **optimal window length** that achieves the mmse.

Batch estimation of the MTF

- The MTF approximation can be written in a vector form as

$$\hat{\mathbf{y}}_k\left(h_k\right) = \mathbf{x}_k\, h_k$$

LS estimate:

$$\hat{h}_k = \arg\min_{h_k} \left\|\mathbf{y}_k - \mathbf{x}_k\, h_k\right\|^2 = \frac{\mathbf{x}_k^H \mathbf{y}_k}{\mathbf{x}_k^H \mathbf{x}_k}$$

- The (normalized) mse is defined by

$$\epsilon = \frac{\sum_{k=0}^{N-1} E\left\{\left\|\mathbf{d}_k - \hat{\mathbf{y}}_k\left(\hat{h}_k\right)\right\|^2\right\}}{\sum_{k=0}^{N-1} E\left\{\left\|\mathbf{d}_k\right\|^2\right\}}$$

Batch estimation of the MTF

- The MTF approximation can be written in a vector form as

$$\hat{\mathbf{y}}_k\left(h_k\right) = \mathbf{x}_k\, h_k$$

LS estimate:

$$\hat{h}_k = \arg\min_{h_k} \left\|\mathbf{y}_k - \mathbf{x}_k\, h_k\right\|^2 = \frac{\mathbf{x}_k^H \mathbf{y}_k}{\mathbf{x}_k^H \mathbf{x}_k}$$

- **The (normalized) mse is defined by**

$$\epsilon = \frac{\sum_{k=0}^{N-1} E\left\{\left\|\mathbf{d}_k - \hat{\mathbf{y}}_k\left(\hat{h}_k\right)\right\|^2\right\}}{\sum_{k=0}^{N-1} E\left\{\left\|\mathbf{d}_k\right\|^2\right\}}$$

MSE analysis

- Let $\eta = \sigma_x^2/\sigma_\xi^2$ denote the SNR.

The mmse obtainable by the MTF approximation:

$$\epsilon = \epsilon_N + \epsilon_P$$

$$\text{where} \quad \epsilon_N = 1 - a \quad \text{and} \quad \epsilon_P = \frac{1}{P}\left(b/\eta - c\right)$$

[a ,b and c depend on $h(n)$ and $\tilde{\psi}(n)$]

[Avargel & Cohen, IEEE Signal Process. Letters, 07']

- ϵ_N is attributable to using a finite-support analysis window $\Longrightarrow \epsilon_N(N \to \infty) = 0$.
- ϵ_P is a consequence of restricting the length of the input signal $\Longrightarrow \epsilon_P(P \to \infty) = 0$.

MSE analysis

- Let $\eta = \sigma_x^2/\sigma_\xi^2$ denote the SNR.

The mmse obtainable by the MTF approximation:

$$\epsilon = \epsilon_N + \epsilon_P$$

$$\text{where} \quad \epsilon_N = 1 - a \quad \text{and} \quad \epsilon_P = \frac{1}{P}\left(b/\eta - c\right)$$

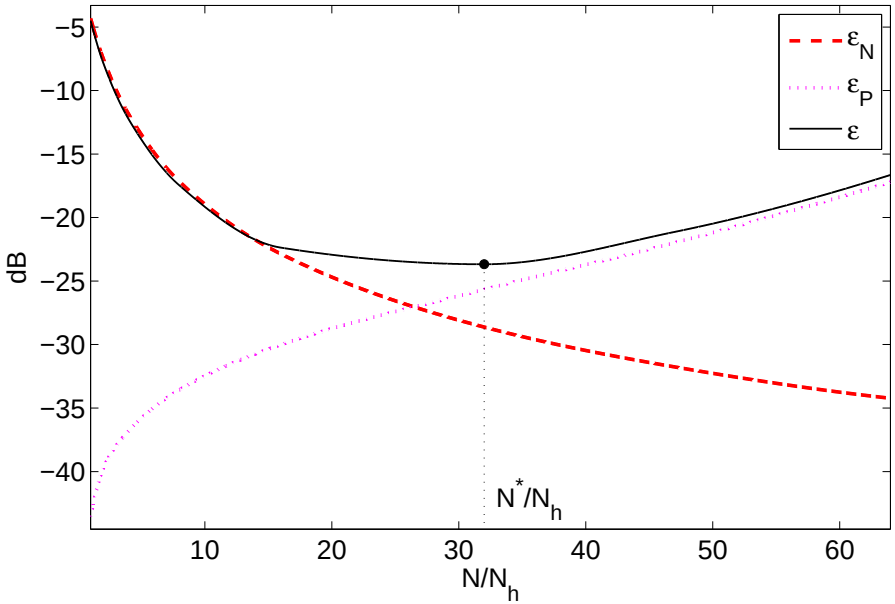
[a ,b and c depend on $h(n)$ and $\tilde{\psi}(n)$]

[Avargel & Cohen, IEEE Signal Process. Letters, 07']

- ϵ_N is attributable to using a finite-support analysis window $\Longrightarrow \epsilon_N(N \to \infty) = 0$.
- ϵ_P is a consequence of restricting the length of the input signal $\Longrightarrow \epsilon_P(P \to \infty) = 0$.

Discussion

Theoretical MSE curves for a 0 dB SNR:



- ϵ_N is a monotonically *decreasing* function of N, while ϵ_P is a monotonically *increasing* function.

Optimal window length

The total mse ϵ may reach its minimum value for a certain optimal window length N^*.

$$N^* = \arg\min_N \epsilon$$

We show that...

As the SNR or the input signal length increases, a longer analysis window should be used to make the MTF approximation valid and the variance of the MTF estimate reasonably low.

Optimal window length

The total mse ϵ may reach its minimum value for a certain optimal window length N^*.

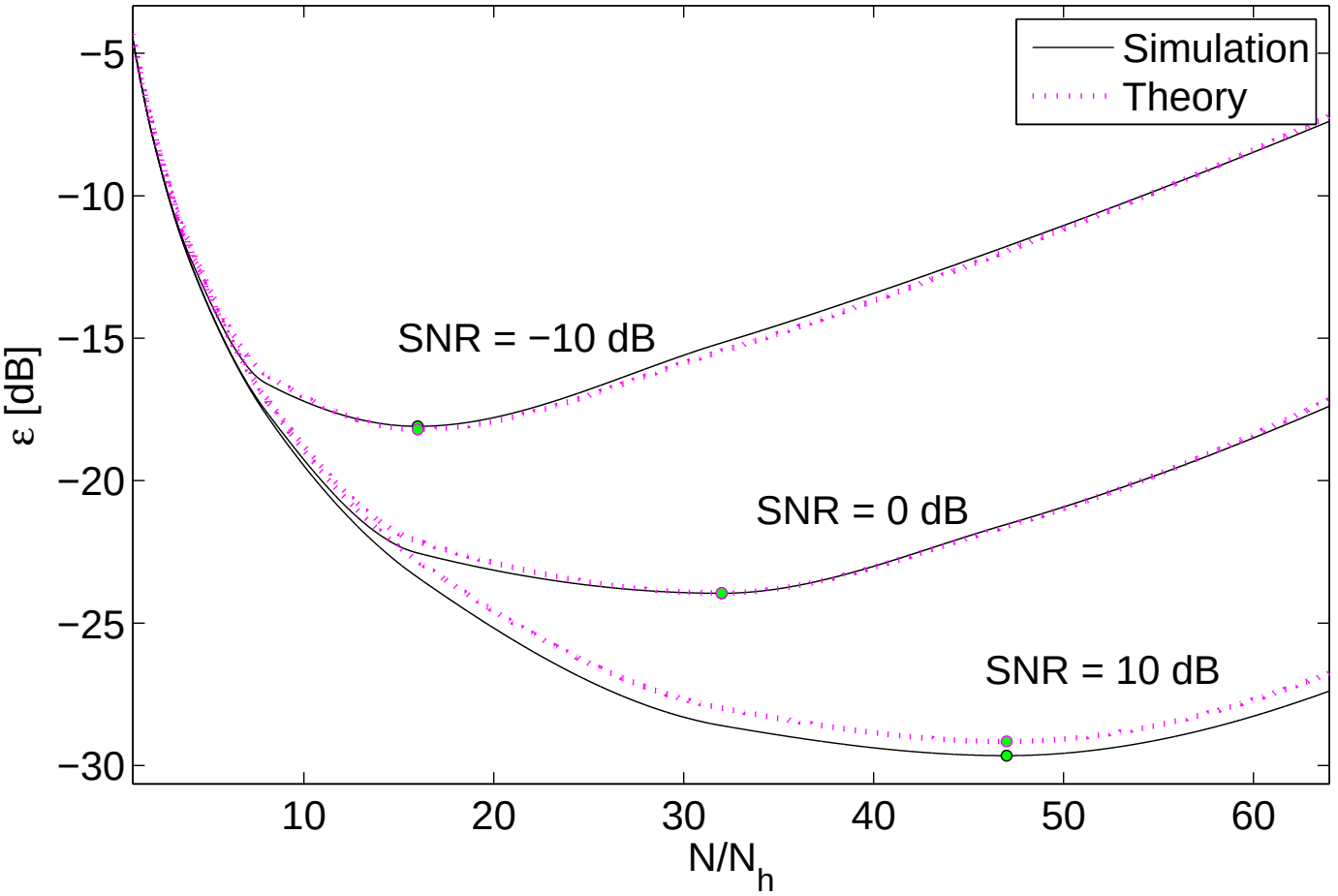
$$N^* = \arg\min_N \epsilon$$

We show that...

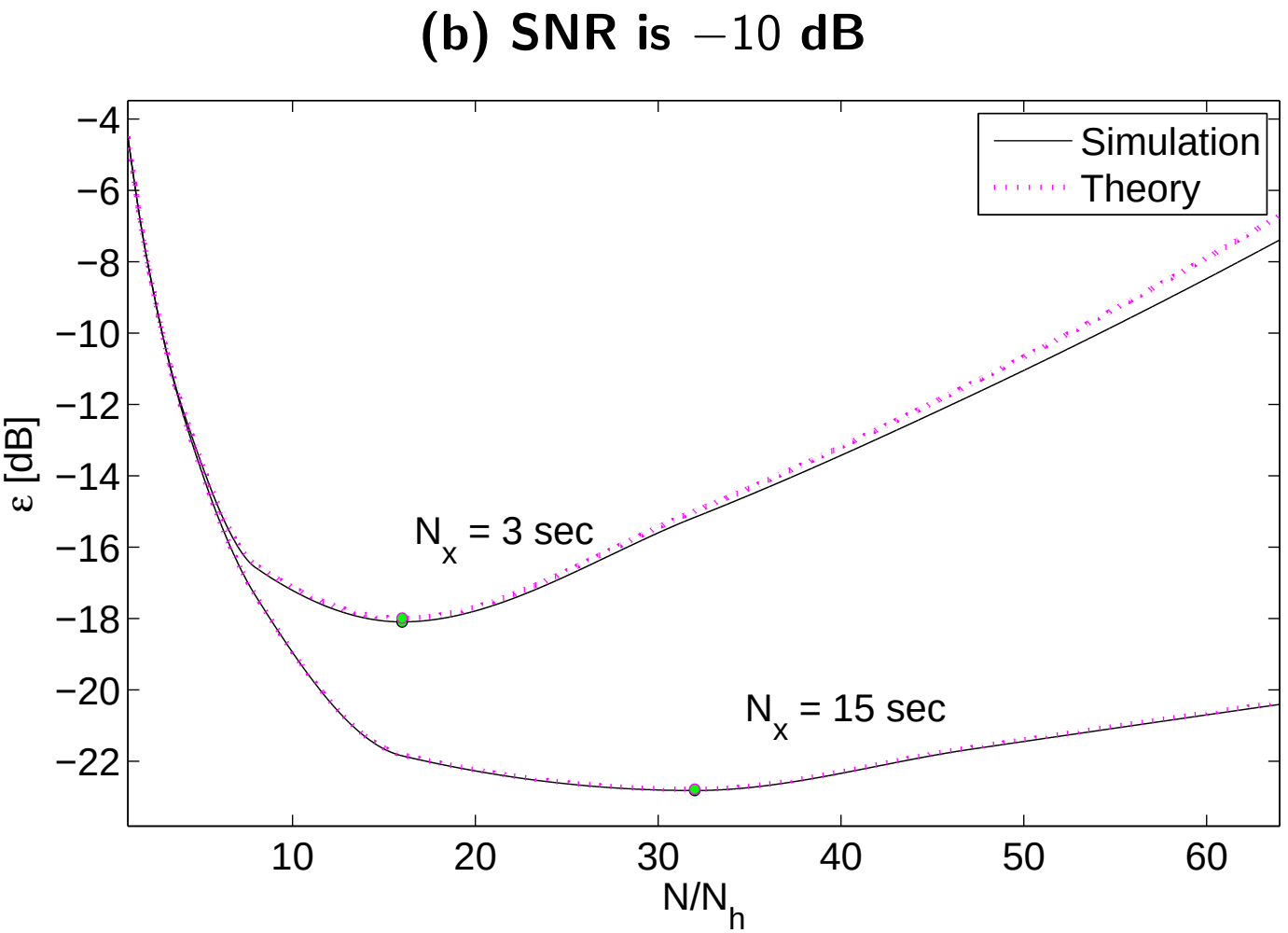
As the SNR or the input signal length increases, a longer analysis window should be used to make the MTF approximation valid and the variance of the MTF estimate reasonably low.

Experimental results

(a) N_x is 3 seconds



Experimental results (cont.)



Adaptive Identification: Cross-MTF

- A new model is proposed to improve the MTF approach.

The cross-MTF approximation:

$$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} h_{k,k'} x_{p,k'}$$

[Avargel & Cohen, IEEE Trans. Audio Speech Lang. Process., 08']

- Estimation of additional cross-terms results in a slower convergence, but improves the steady-state mse.

We propose a new algorithm that adaptively controls the number of cross-terms to achieve the mmse at each iteration.

Adaptive Identification: Cross-MTF

- A new model is proposed to improve the MTF approach.

The cross-MTF approximation:

$$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} h_{k,k'} x_{p,k'}$$

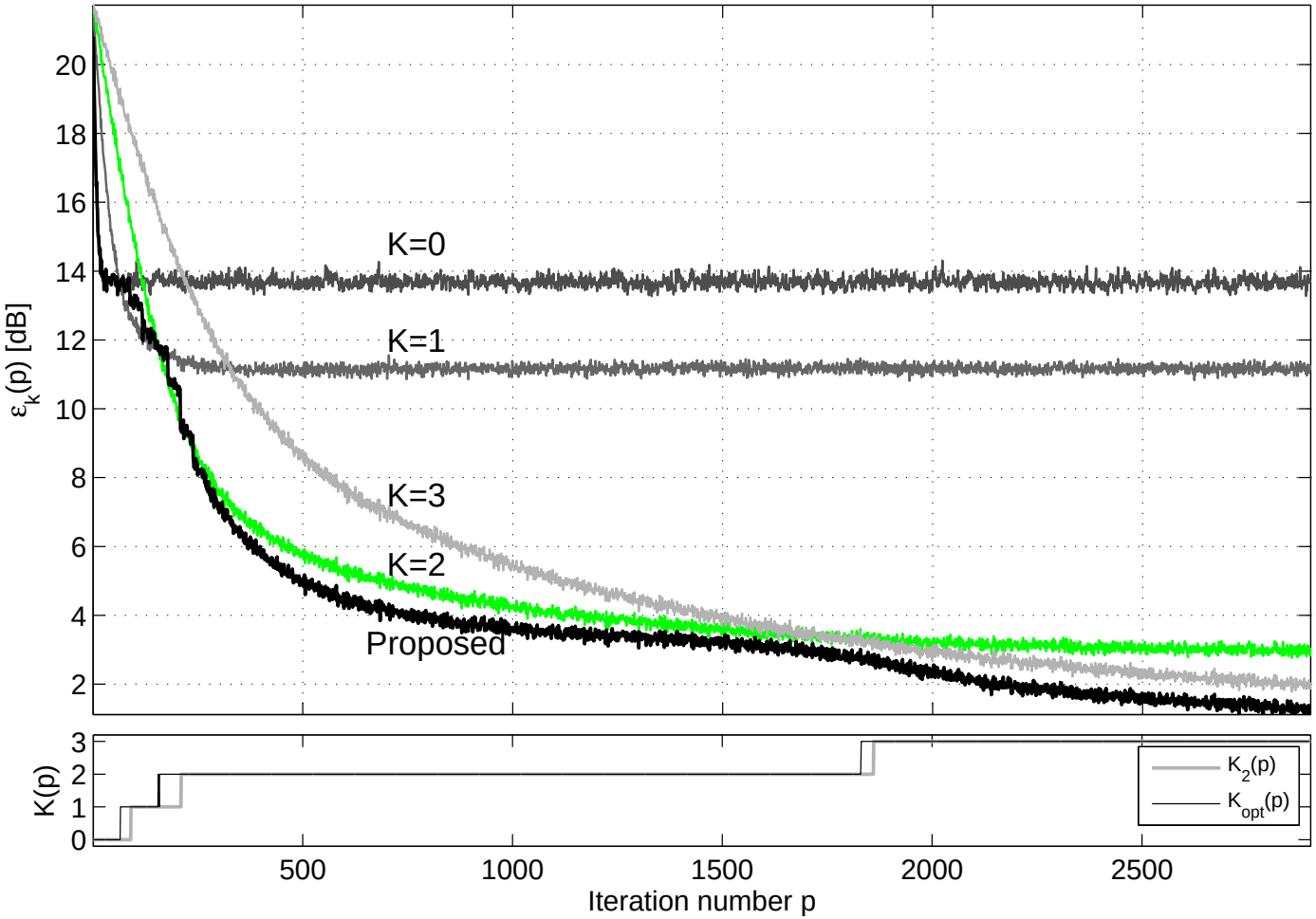
[Avargel & Cohen, IEEE Trans. Audio Speech Lang. Process., 08']

- Estimation of additional cross-terms results in a slower convergence, but improves the steady-state mse.

We propose a new algorithm that adaptively controls the number of cross-terms to achieve the mmse at each iteration.

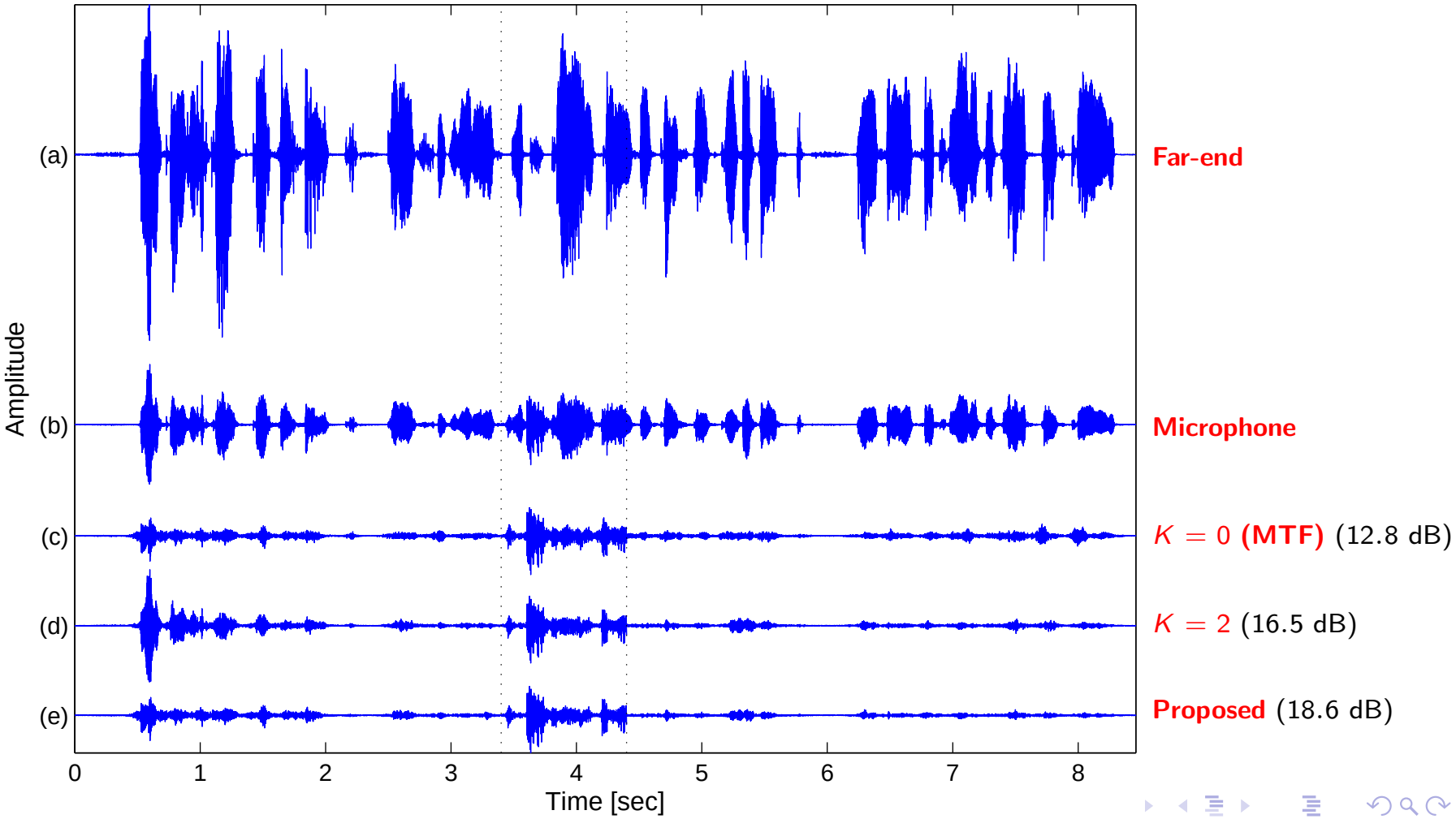
Experimental results (adaptive control)

White Gaussian signals



Experimental results

Acoustic echo cancellation application:



Nonlinear Systems in the STFT Domain

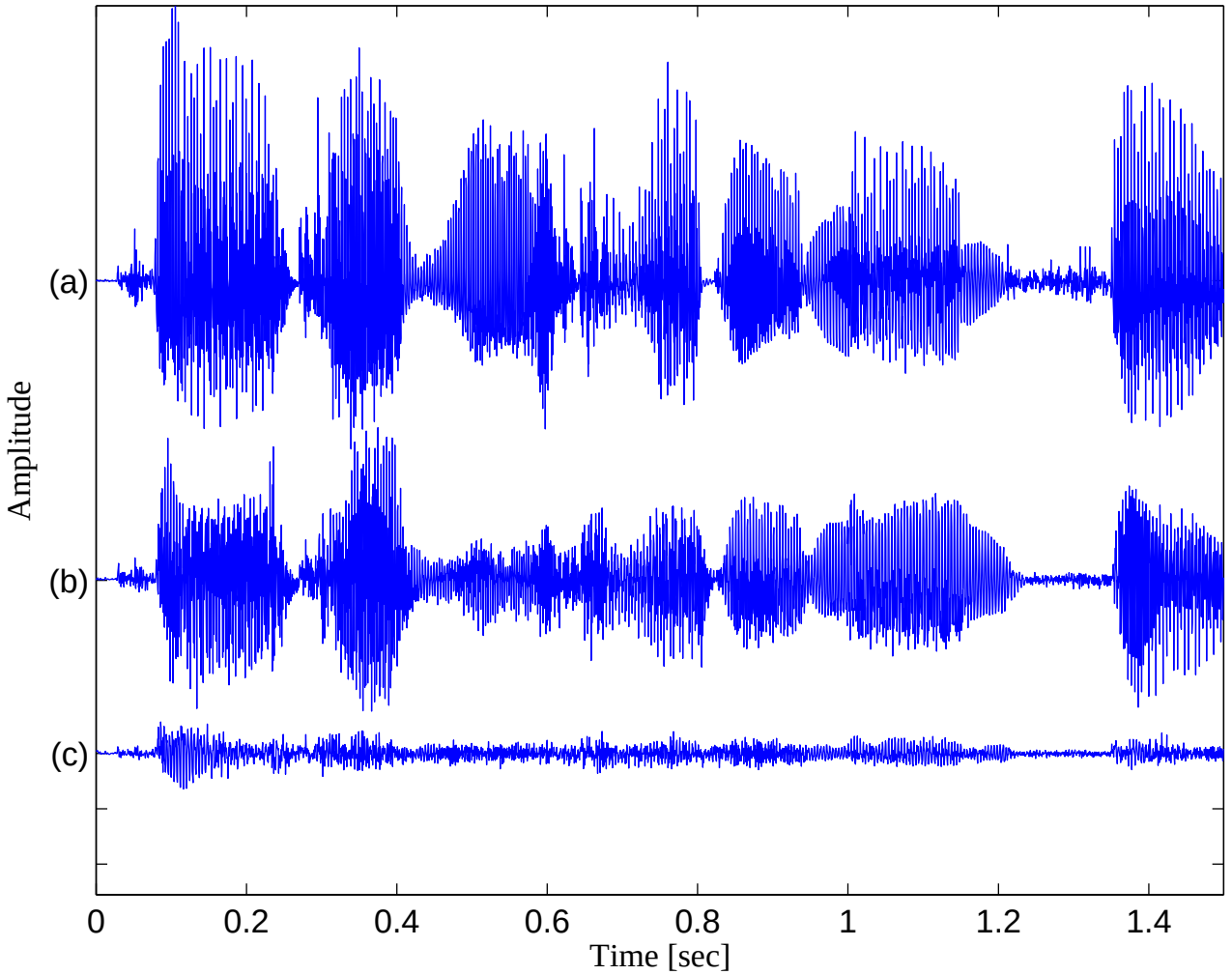
Nonlinear system identification

- So far, **linear** models have been considered:

In many real-world applications, the considered systems exhibit certain nonlinearities that cannot be sufficiently estimated by conventional linear models.

- In acoustic echo cancellation applications, nonlinearities are introduced by the loudspeakers and their amplifiers.

Nonlinear acoustic echo cancellation



(a). Loudspeaker

(b). Microphone

(c). Linear (14.5 dB)

Volterra filters

- A popular and widely-used nonlinear system representation is the **Volterra filter**.
- The Volterra series was developed in 1887 by Vito Volterra.

- A Volterra series denotes a nonlinear time-invariant operation, which can be regarded as a Taylor series expansion with memory.

Volterra filters (cont.)

qth-order Volterra filter:

$$d(n) = \sum_{\ell=1}^{q} d_\ell(n)$$

$$d_\ell(n) = \sum_{m_1=0}^{N_\ell-1} \cdots \sum_{m_\ell=0}^{N_\ell-1} h_\ell(m_1, \ldots m_\ell) \prod_{i=1}^{\ell} x(n-m_i)$$

where $d_\ell(n)$ denotes the ℓth-order homogeneous Volterra filter, and $h_\ell(m_1, \ldots m_\ell)$ is the ℓ**th-order Volterra kernel**.

- N_ℓ represents the memory length of each kernel.

Existing Approaches

Nonlinear system identification using Volterra filters aims at estimating the Volterra kernels based on input output data:

- Noisy observations: $y(n) = d(n) + \xi(n)$.

- Volterra-filter estimation methods are divided into two groups:
  - Time-domain approaches - aims at estimating the Volterra kernels.
  - Frequency-domain approaches - aims at estimating the Volterra transfer functions.

Existing Approaches

Nonlinear system identification using Volterra filters aims at estimating the Volterra kernels based on input output data:

- Noisy observations: $y(n) = d(n) + \xi(n)$.

- Volterra-filter estimation methods are divided into two groups:
  - Time-domain approaches - aims at estimating the Volterra kernels.
  - Frequency-domain approaches - aims at estimating the Volterra transfer functions.

Time domain approaches

- Linear dependency: The Volterra filter output is given by

$$d(n) = \mathbf{h}^T \mathbf{x}(n)$$

where $\mathbf{h}$ consists of the Volterra kernels, and $\mathbf{x}(n)$ is the corresponding input vector.

- Linear **batch** methods and **adaptive** filtering algorithms are traditionally used.

Batch estimation [Ljung, 78'], [Nowak, 98'], [Glentis, 99']

$$\hat{\mathbf{h}}_{MSE} = \left[E\left\{ \mathbf{x}^T(n)\mathbf{x}(n) \right\} \right]^{-1} E\left\{ \mathbf{x}(n) y(n) \right\}$$

$$\hat{\mathbf{h}}_{LS} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}\mathbf{y}$$

Time domain approaches (cont.)

Adaptive estimation [Kou & Powers, 85'], [Glentis, 99'], [Guerin, 03']

LMS: $$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu \mathbf{x}(n) \left[y(n) - \hat{\mathbf{h}}(n)^T \mathbf{x}(n) \right]$$

- **Drawbacks:**
  - The Volterra model suffer from severe **ill-conditioning** $\Longrightarrow$ difficult to estimate from short/noisy data.
  - Extremely **high computational cost** for nonlinear systems with large memory length. Number of parameters:

$$\sum_{\ell=1}^{Q} \begin{pmatrix} N_\ell + \ell - 1 \\ \ell \end{pmatrix}$$

  - **Slow convergence** of adaptive Volterra filters due to the large number of parameters and the correlated input vector.

STFT representation of nonlinear systems

Main goal:

To introduce a new nonlinear model in the STFT domain for improved nonlinear system identification.

Why should nonlinear systems be modeled in the STFT domain?

Computational cost may be reduced due to the decimation factor of the STFT $\Longrightarrow$ nonlinear system with large memory length can be estimated.

An STFT-based nonlinear model may be **combined** with efficient algorithms already implemented in the STFT domain,

e.g., crossband filters for the linear kernel representation.

STFT representation of nonlinear systems

Main goal:

To introduce a new nonlinear model in the STFT domain for improved nonlinear system identification.

Why should nonlinear systems be modeled in the STFT domain?

Computational cost may be reduced due to the decimation factor of the STFT $\Longrightarrow$ nonlinear system with large memory length can be estimated.

An STFT-based nonlinear model may be **combined** with efficient algorithms already implemented in the STFT domain,

e.g., crossband filters for the linear kernel representation.

STFT representation of nonlinear systems

Main goal:

To introduce a new nonlinear model in the STFT domain for improved nonlinear system identification.

Why should nonlinear systems be modeled in the STFT domain?

Computational cost may be reduced due to the decimation factor of the STFT $\Longrightarrow$ nonlinear system with large memory length can be estimated.

An STFT-based nonlinear model may be **combined** with efficient algorithms already implemented in the STFT domain,

e.g., crossband filters for the linear kernel representation.

Volterra representation in the STFT domain

- Without loss of generality, the **quadratic** case is considered.
- **A second-order Volterra filter:**

$$\begin{aligned} d(n) &= \sum_{m=0}^{N_1-1} h_1(m)x(n-m) \\ &\quad + \sum_{m=0}^{N_2-1}\sum_{\ell=0}^{N_2-1} h_2(m,\ell)x(n-m)x(n-\ell) \\ &\triangleq d_1(n) + d_2(n) \end{aligned}$$

$h_1(m)$ and $h_2(m,\ell)$ are the *linear* and *quadratic* Volterra kernels, respectively.

Volterra representation in the STFT domain (cont.)

- Applying the STFT to $d_2(n)$ we obtain

$$d_{2;p,k} = \sum_{k',k''=0}^{N-1} \sum_{p',p''} x_{p-p',k'} x_{p-p'',k''} c_{p',p'',k,k',k''}$$

[Avargel & Cohen, submitted to IEEE Trans. Signal Process.]

- For a given frequency-bin index k, the temporal signal $d_{2;p,k}$ consists of all possible combinations of input frequencies taken two at a time.
- The contribution of each frequency couple $\{k', k'' | \, k', k'' \in \{0, \ldots, N-1\}\}$ to the output signal at frequency bin k is given as a **Volterra-like expansion** with $c_{p',p'',k,k',k''}$ being its quadratic kernel.

Volterra representation in the STFT domain (cont.)

- Applying the STFT to $d_2(n)$ we obtain

$$d_{2;p,k} = \sum_{k',k''=0}^{N-1} \sum_{p',p''} X_{p-p',k'} X_{p-p'',k''} C_{p',p'',k,k',k''}$$

[Avargel & Cohen, submitted to IEEE Trans. Signal Process.]

- For a given frequency-bin index k, the temporal signal $d_{2;p,k}$ consists of all possible combinations of input frequencies taken two at a time.
- The contribution of each frequency couple $\{k', k'' | \, k', k'' \in \{0, \ldots, N-1\}\}$ to the output signal at frequency bin k is given as a **Volterra-like expansion** with $C_{p',p'',k,k',k''}$ being its quadratic kernel.

Volterra representation in the STFT domain (cont.)

- The kernel $c_{p',p'',k,k',k''}$ is given by

$$c_{p',p'',k,k',k''} = \left\{ h_2(n,m) * \phi_{k,k',k''}(n,m) \right\} \Big|_{n=p'L,\ m=p''L}$$

- The DTFT of $\phi_{k,k',k''}(n,m)$ is

$$\Phi_{k,k',k''}(\omega,\eta) = \tilde{\Psi}^* \left(\omega + \eta - \frac{2\pi}{N} k \right) \Psi \left(\omega - \frac{2\pi}{N} k' \right) \Psi \left(\omega - \frac{2\pi}{N} k'' \right)$$

Volterra representation in the STFT domain (cont.)

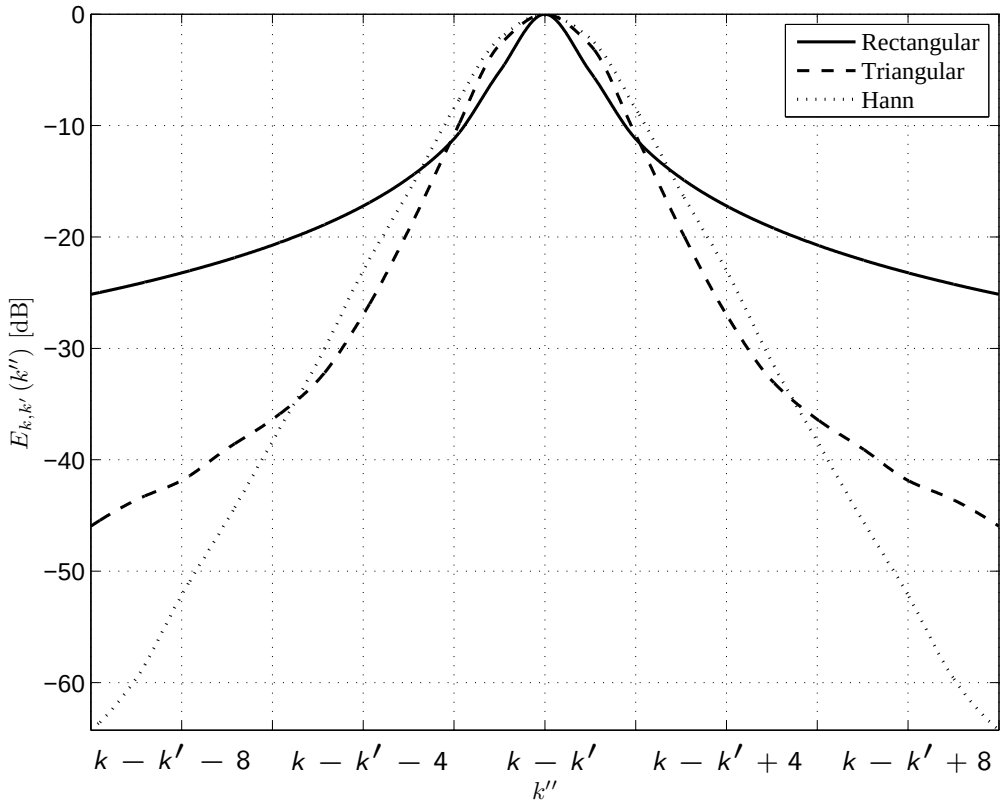


Figure: The energy of $\phi_{k,k',k''}(n,m)$ for $k = 1$ and $k' = 0$, as obtained for different synthesis windows of length $N = 256$.

An approximate model

- To reduce the model complexity, let us assume that the analysis and synthesis filters are selective enough with bandwidths of nearly π/N.
- Accordingly, most of the energy of $c_{p',p'',k,k',k''}$ is concentrated in a small region around the index $k'' = (k - k') \bmod N$, such that

$$d_{2;p,k} \approx \sum_{k'=0}^{N-1} \sum_{p',p''} x_{p-p',k'} x_{p-p'',(k-k') \bmod N} c_{p',p'',k,k',(k-k') \bmod N}$$

An approximate model (cont.)

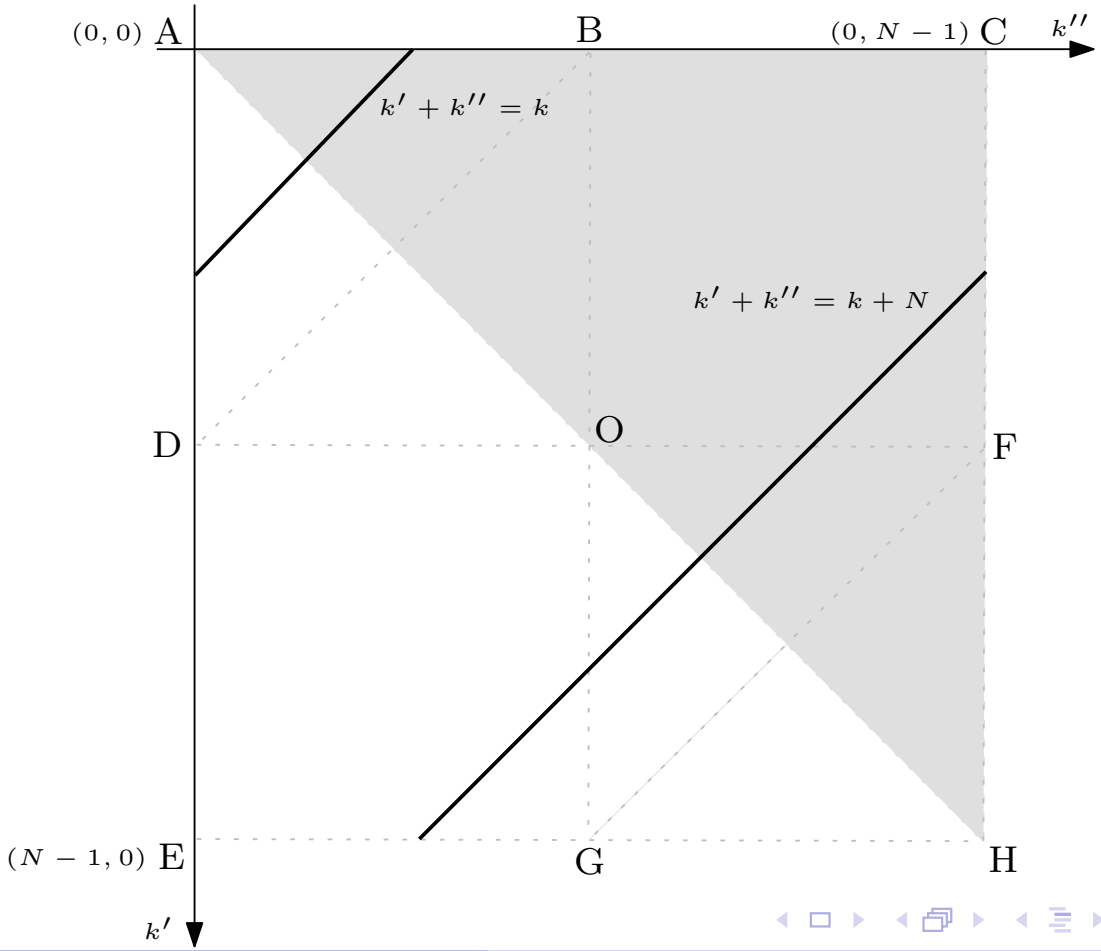
- Extending the so-called cross-multiplicative transfer function (CMTF) approximation to this case, a kernel $c_{p',p'',k,k',k''}$ may be approximated as purely multiplicative in the STFT domain:

$$d_{2;p,k} \approx \sum_{k'=0}^{N-1} x_{p,k'} x_{p,(k-k') \bmod N} c_{k',(k-k') \bmod N}$$

- We refer to $c_{k',k''}$ as a **quadratic cross-term**.

An approximate model (cont.)

- Only frequency indices $\{k', k''\}$, whose sum is k or $k + N$, contribute to the output at frequency bin k.



An approximate model (cont.)

- **Finally, the proposed model for quadratically nonlinear systems in the STFT domain is given by:**

$$d_{p,k} = \sum_{k'=0}^{N-1} \sum_{p'=0}^{M-1} x_{p-p',k'} h_{p',k,k'} + \sum_{k'=0}^{N-1} x_{p,k'} x_{p,(k-k') \bmod N} c_{k',(k-k') \bmod N}$$

[Avargel & Cohen, submitted to IEEE Trans. Signal Process.]

where $h_{p',k,k'}$ is a crossband filters, and $c_{k',(k-k') \bmod N}$ is a quadratic cross-term.

An approximate model (cont.)

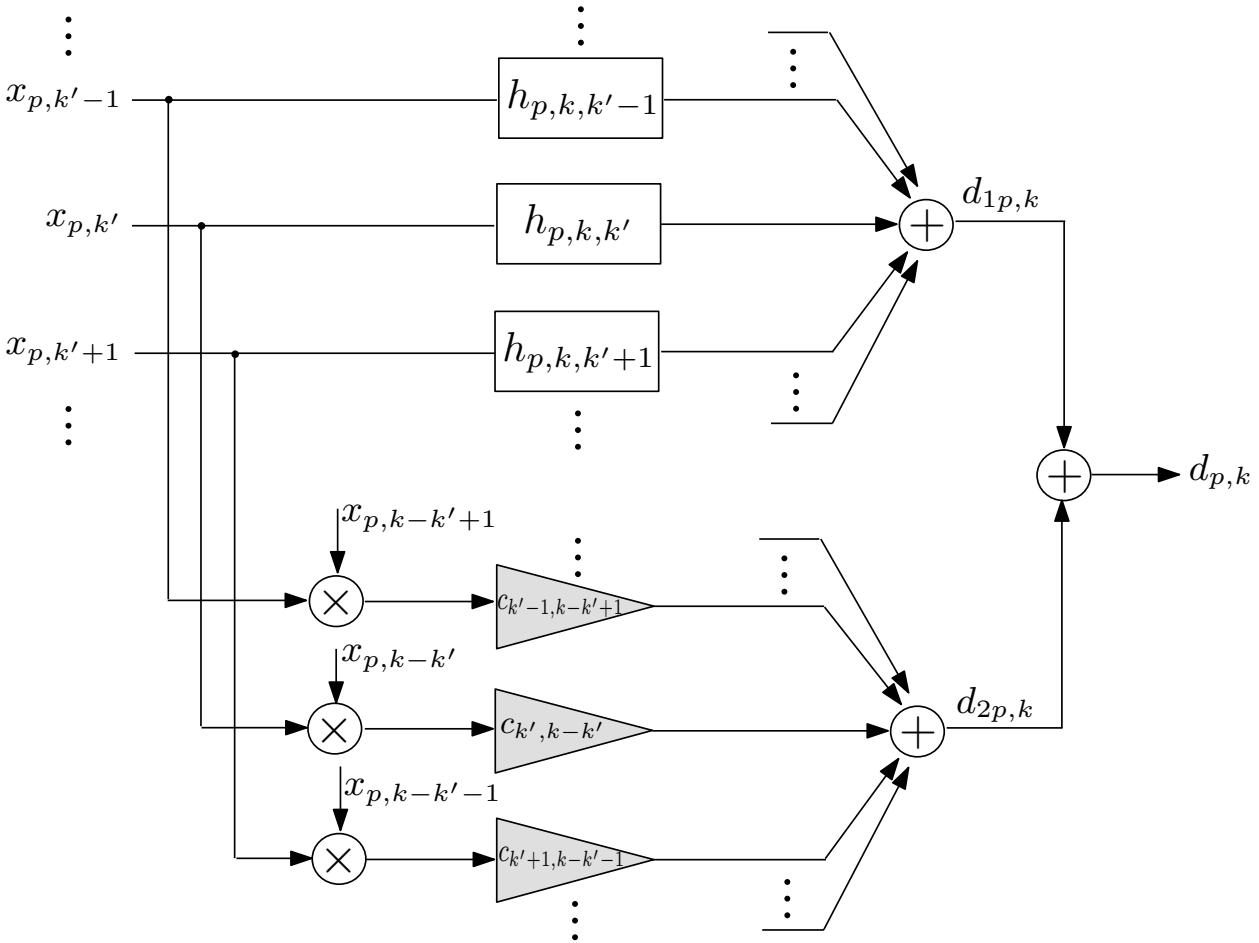


Figure: Block diagram of the proposed model.

Batch estimation

- Employing the proposed quadratic STFT model, an estimator for the system output in the STFT domain can be written as

$$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} \sum_{p'=0}^{M-1} x_{p-p',k' \bmod N} h_{p',k,k' \bmod N} + \sum_{k'=0}^{N-1} x_{p,k'} x_{p,(k-k') \bmod N} c_{k',(k-k') \bmod N}$$

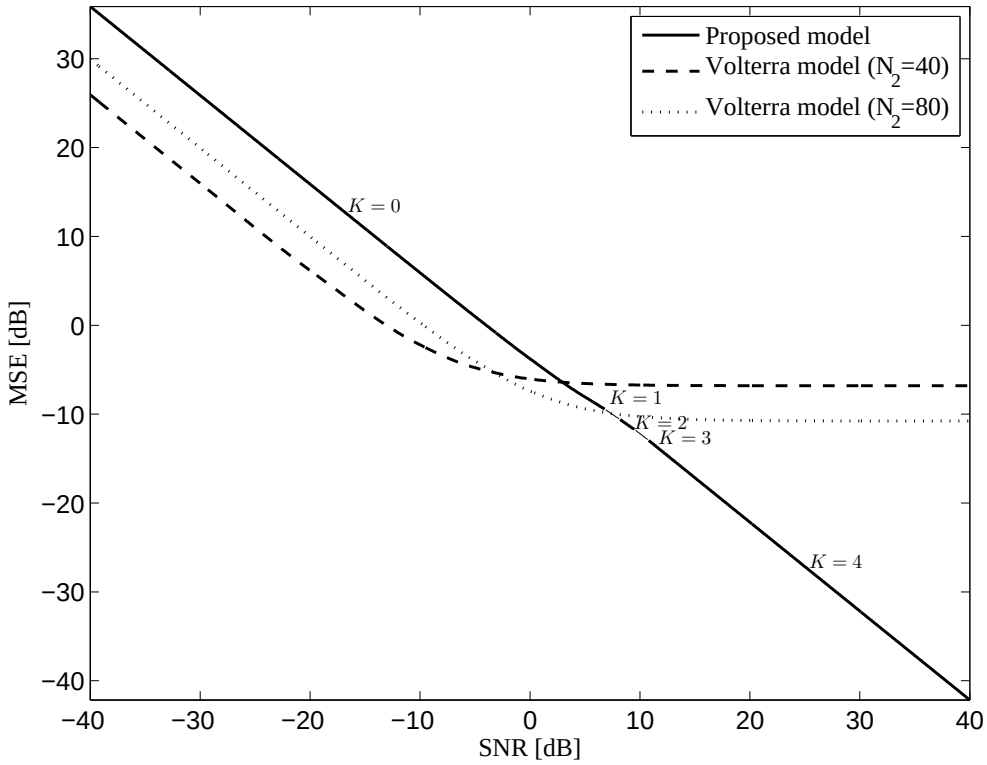
LS estimate:

$$\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}_k} \|\mathbf{y}_k - \mathbf{R}_k \boldsymbol{\theta}_k\|^2$$

Experimental results

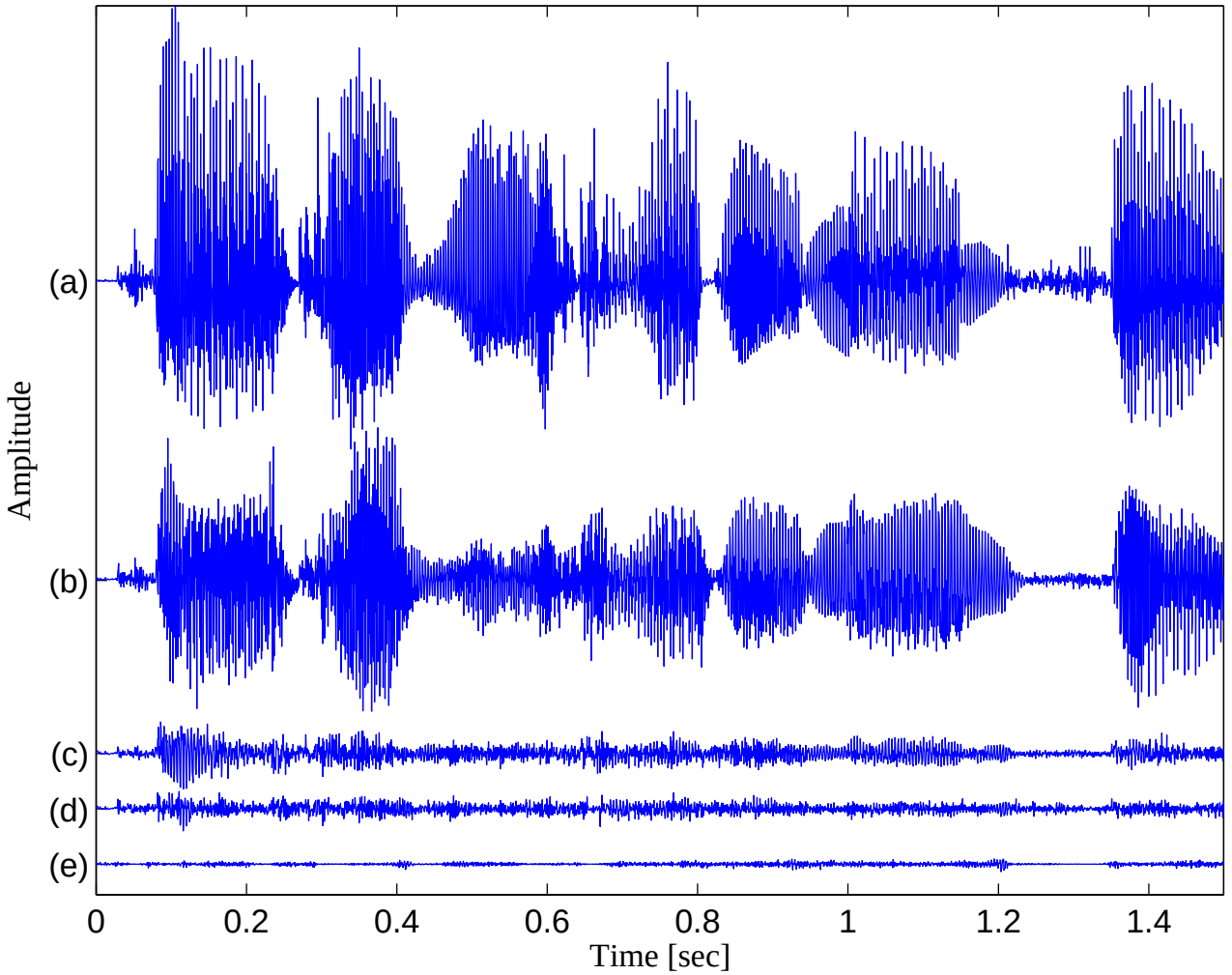
White Gaussian input signals:

$$y(n) = \sum_{m=0}^{767} g_1(m)x(n-m) + \sum_{m=0}^{767} g_1(m)x^2(n-m) + \xi(n)$$



Experimental results (cont.)

Acoustic echo cancellation application:



(a). Loudspeaker

(b). Microphone

(c). Linear (14.5 dB)

(d). Volterra (19.1 dB)

(e). Proposed (29.5 dB)

Computational complexity

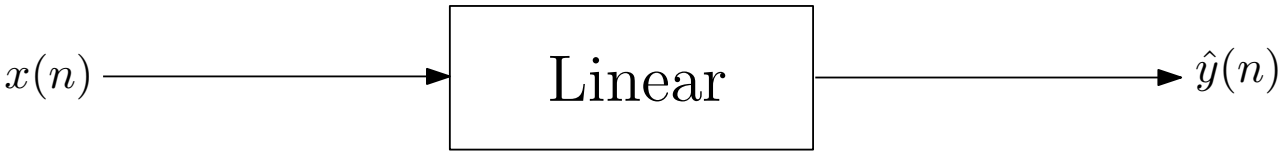
- Let $r = L/N$. Then, the ratio between the Volterra (fullband) and proposed (subband) complexities is given by

$$\frac{O_f}{O_s} \sim r \frac{\left(2N_1 + N_2^2\right)^2}{\left[2N_1 \frac{(2K+1)}{rN} + N\right]^2}$$

- For instance, for $N = 256$, $r = 0.5$ (i.e., $L = 128$), $N_1 = 1024$, $N_2 = 80$ and $K = 2$ the proposed approach complexity is reduced by approximately 300.
- Computational efficiency obtained by the proposed approach becomes even more significant when systems with **long memory** are considered.

Nonlinear undermodeling error

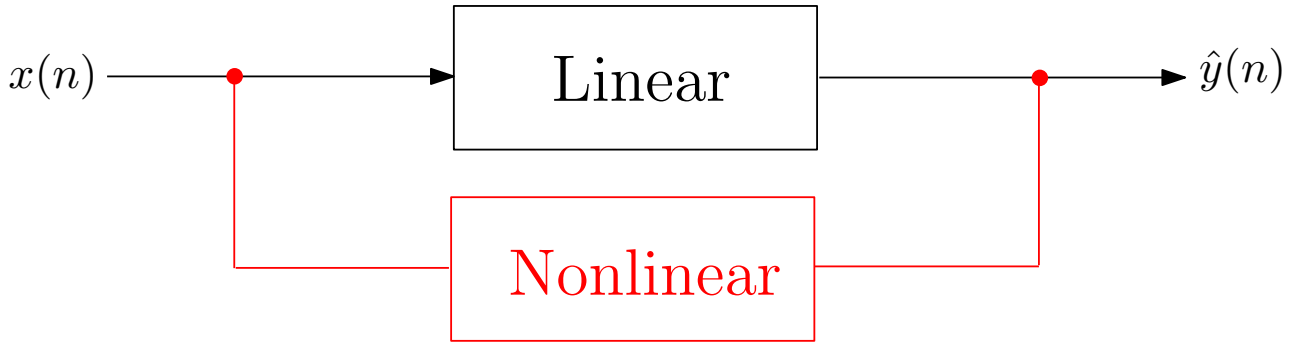
Is the inclusion of a nonlinear component in the model always preferable?



- Employing a purely linear model for nonlinear system estimation is referred to as **nonlinear undermodeling**.
- Quantifying the nonlinear undermodeling error is of major importance since in many cases a purely linear model is fitted to the data.

Nonlinear undermodeling error

Is the inclusion of a nonlinear component in the model always preferable?



- Employing a purely linear model for nonlinear system estimation is referred to as **nonlinear undermodeling**.
- Quantifying the nonlinear undermodeling error is of major importance since in many cases a purely linear model is fitted to the data.

Nonlinear undermodeling error (cont.)

- We investigate the influence of nonlinear undermodeling in the STFT domain for batch and adaptive estimation schemes, taking into account:
  - Noise level (SNR).
  - Data length.
  - Power ratio of nonlinear to linear components (NLR).

$$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} \sum_{p'=0}^{M-1} x_{p-p',k' \bmod N} h_{p',k,k' \bmod N} + \gamma \sum_{k'=0}^{N-1} x_{p,k'} x_{p,(k-k') \bmod N} c_{k',(k-k') \bmod N}$$

$\gamma \in \{0,1\}$ determines whether the nonlinear component is included in the model structure.

Nonlinear undermodeling error (cont.)

- We investigate the influence of nonlinear undermodeling in the STFT domain for batch and adaptive estimation schemes, taking into account:
  - Noise level (SNR).
  - Data length.
  - Power ratio of nonlinear to linear components (NLR).

$$\hat{y}_{p,k} = \sum_{k'=k-K}^{k+K} \sum_{p'=0}^{M-1} x_{p-p',k' \bmod N} h_{p',k,k' \bmod N} + \gamma \sum_{k'=0}^{N-1} x_{p,k'} x_{p,(k-k') \bmod N} c_{k',(k-k') \bmod N}$$

$\gamma \in \{0, 1\}$ determines whether the nonlinear component is included in the model structure.

Batch estimation

The model parameters are estimated off-line:

LS estimate:

$$\hat{\boldsymbol{\theta}}_{\gamma k} = \arg\min_{\boldsymbol{\theta}_k} \|\mathbf{y}_k - \mathbf{R}_{\gamma k}\boldsymbol{\theta}_k\|^2$$
$$= \left(\mathbf{R}_{\gamma k}^H \mathbf{R}_{\gamma k}\right)^{-1} \mathbf{R}_{\gamma k}^H \mathbf{y}_k$$

$\epsilon_{0k}(K)$ - the mse obtained by using only a **linear** model.

$\epsilon_{1k}(K)$ - the mse obtained by incorporating also a **quadratic** component into the model .

MSE analysis

We derive an explicit expression for the mmse:

$$\epsilon_{\gamma k}(K) = \frac{\alpha_{\gamma k}(K)}{\eta} + \beta_{\gamma k}(K)$$

$$\alpha_{\gamma k}(K) \triangleq \frac{(2K+1)\,M}{P} + \gamma \frac{N/2+1}{P}$$

$$\beta_{\gamma k}(K) \triangleq 1 - \frac{(2K+1)\,M}{P} - \|\mathbf{h}_k\|^{-2} \left[h_1(K) + \frac{\sigma_x^2 c(K)}{P} \right] \frac{1}{1+\varphi}$$

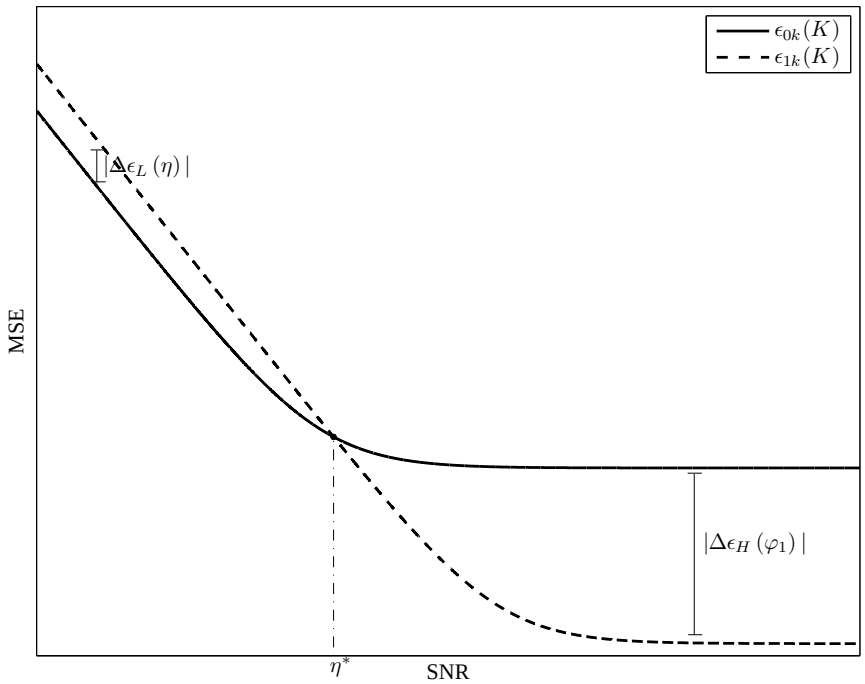
$$-\gamma \left[\frac{1 + N/2 + \|\mathbf{h}_k\|^{-2} h_2}{P} + \varphi \right] \frac{1}{1+\varphi}$$

$$\eta = \sigma_d^2/\sigma_\xi^2 \text{ (SNR)} \; ; \; \varphi = \sigma_{d_Q}^2/\sigma_{d_L}^2 \text{ (NLR)} \; ;$$

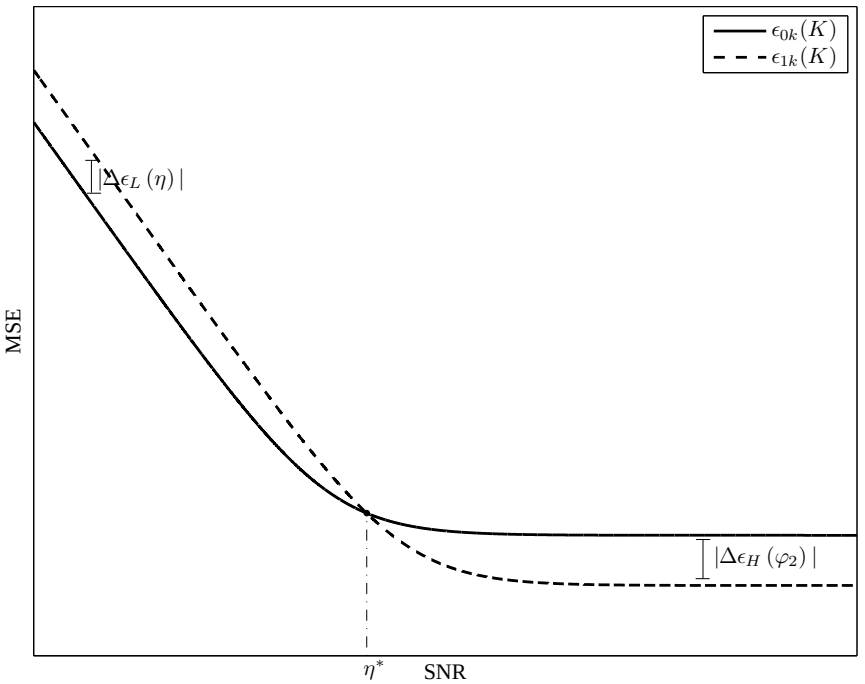
$$h_1(K) \triangleq \sum_{m=0}^{2K} \left\| \bar{\mathbf{h}}_{k,(k-K+m) \bmod N} \right\|^2 \; ; \; h_2 \triangleq \sum_{k'=0}^{N-1} \left| \bar{h}_{0,k,k'} \right|^2 \; ; \; c(K) \triangleq \sum_{i=1}^{4} \sum_{m \in \mathcal{L}_i} \left| \bar{c}_{m,(k-m) \bmod N} \right|^2 .$$

Theoretical MSE curves

What can be verified from the MSE expression?



(a) NLR $= \varphi_1$



(b) NLR $= 0.2\varphi_1$

Discussion

- $\epsilon_{1k}(K) > \epsilon_{0k}(K)$ for low SNR ($\eta << 1$), and $\epsilon_{1k}(K) \leq \epsilon_{0k}(K)$ for high SNR ($\eta >> 1$) $\Longrightarrow$ **as the SNR increases, the mse performance can be generally improved by incorporating also the nonlinear component into the model ($\gamma = 1$).**

- The stronger the nonlinearity of the system, the larger the improvement achieved by using the full nonlinear model.

- As the nonlinearity becomes weaker, higher SNR should be considered to justify the inclusion of the nonlinear component.

Adaptive estimation

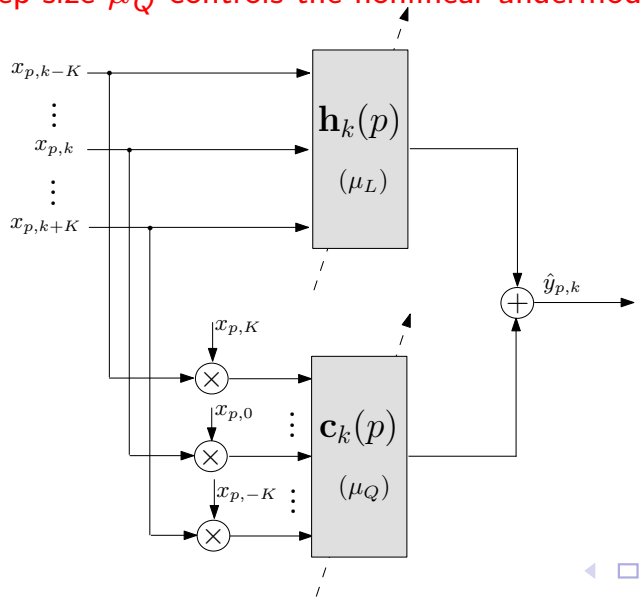
- The nonlinear model parameters are adaptively estimated:

LMS adaptation:

$$\mathbf{h}_k(p+1) = \mathbf{h}_k(p) + \mu_L e_{p,k} \mathbf{x}^*_{Lk}(p)$$

$$\mathbf{c}_k(p+1) = \mathbf{c}_k(p) + \mu_Q e_{p,k} \mathbf{x}^*_{Qk}(p)$$

The step-size μ_Q controls the nonlinear undermodeling



MSE analysis

Transient performance:

$$\epsilon_k(p) = \epsilon_k^{\min} + \sigma_x^2 E\left\{\left\|\mathbf{g}_{Lk}(p)\right\|^2\right\} + \sigma_x^4 E\left\{\left\|\mathbf{g}_{Qk}(p)\right\|^2\right\}$$

$$E\left\{\left\|\mathbf{g}_{Lk}(p+1)\right\|^2\right\} = \alpha_L\, E\left\{\left\|\mathbf{g}_{Lk}(p)\right\|^2\right\} + \beta_L\, E\left\{\left\|\mathbf{g}_{Qk}(p)\right\|^2\right\} + \gamma_L$$

$$E\left\{\left\|\mathbf{g}_{Qk}(p+1)\right\|^2\right\} = \alpha_Q\, E\left\{\left\|\mathbf{g}_{Qk}(p)\right\|^2\right\} + \beta_Q\, E\left\{\left\|\mathbf{g}_{Lk}(p)\right\|^2\right\} + \gamma_Q$$

$\epsilon_k^{\min} = \sigma_\xi^2 + \sigma_x^2 \left\|\tilde{\mathbf{h}}_k\right\|^2$ - *minimum mse*

$\mathbf{g}_{Lk}(p) = \mathbf{h}_k(p) - \bar{\mathbf{h}}_k$ and $\mathbf{g}_{Qk}(p) = \mathbf{c}_k(p) - \bar{\mathbf{c}}_k$ - *misalignment vectors*

The parameters $\alpha_L, \beta_L, \gamma_L, \alpha_Q, \beta_Q, \gamma_Q$ depend on μ_L and μ_Q.

MSE analysis (cont.)

Steady-state performance:

- **Steady-state mse:**

$$\epsilon_k(\infty) = f\left(\mu_L, \mu_Q\right) \epsilon_k^{\min}$$

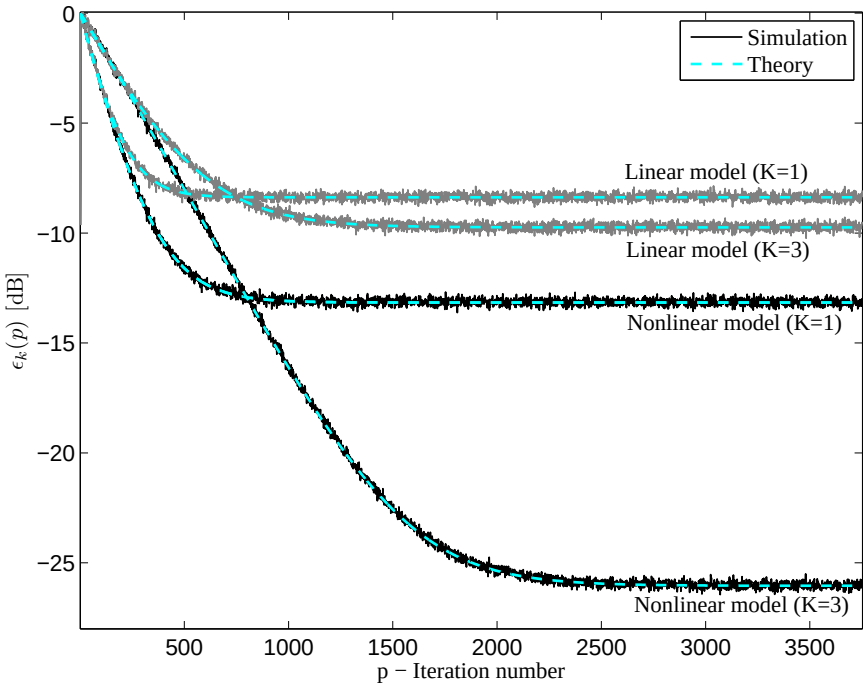
$$f\left(\mu_L, \mu_Q\right) = \frac{2}{2 - \mu_L \sigma_x^2 (2K+1) M - \mu_Q \sigma_x^4 N/2}$$

- **Convergence conditions:**

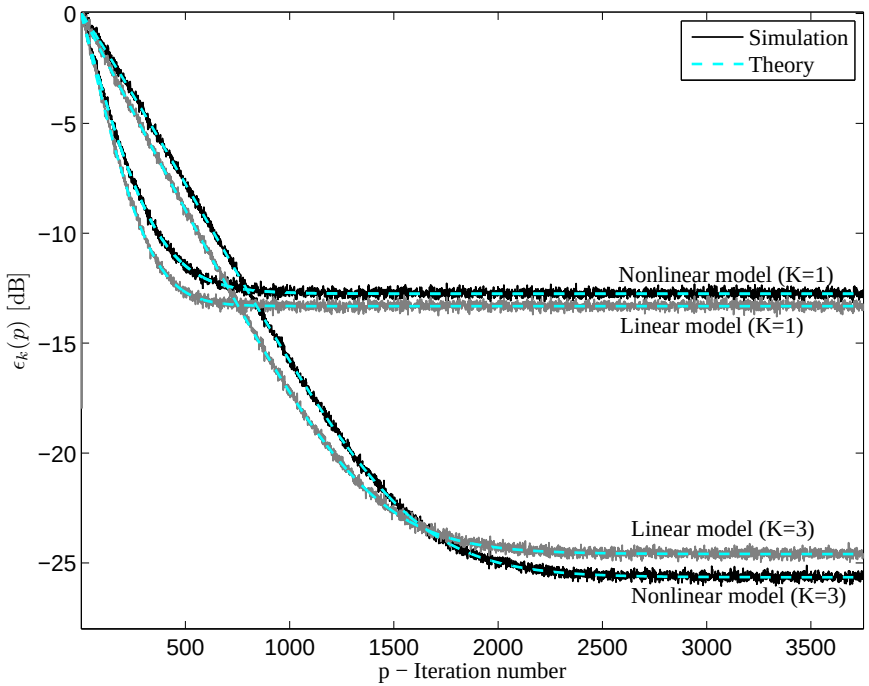
$$0 < \mu_L < \frac{2}{\sigma_x^2 (2K+1) M}$$

$$0 < \mu_Q < \frac{2}{\sigma_x^4 N/2}$$

Results



(a) NLR $= -10$ dB



(b) NLR $= -30$ dB

Discussion

Incorporating the nonlinear component into the model may not necessarily imply a lower steady-state mse in subbands.

- The estimation of the nonlinear component improves the mse performance only when the NLR is relatively high.
- As the nonlinearity becomes weaker, the steady-state mse associated with the linear model decreases, while the relative improvement achieved by the nonlinear model becomes smaller.

Summary

- The problem of linear and nonlinear system identification in the STFT domain has been considered.
- The influence of the system parameters on the model order and model structure has been investigated.
- A novel approach for improved nonlinear system identification has been introduced.

Future Research:

- Adaptive-control algorithms for nonlinear system identification.
- Time-varying system identification in the STFT domain - new models and estimation approaches.
- Extension to multichannel processing (e.g., RTF identification).

Thank you!